Mathematics of Deep Learning

Part III
Information Theory and Learning

Silvano Salvador

Mathematics of Deep Learning Series

Second Edition

Every deep network is simultaneously a compressor and a predictor. The tension between these two roles — how much of the input must be retained to make accurate predictions — is the central question of this volume.

Preface to the Second Edition

This volume constitutes Part III of the Mathematics of Deep Learning series. It develops the information-theoretic foundations of representation learning, generalisation, and generative modelling in a unified mathematical framework. The reader is assumed to have familiarity with probability theory, linear algebra, and the basics of machine learning at the level of Parts I and II.

The second edition expands substantially on the first. Three new chapters have been added: Chapter 9 on connections between information theory and differential privacy; Chapter 10 on the information geometry of neural network training dynamics; and Chapter 11 on federated learning and communication-constrained inference. Every existing chapter has been significantly extended with new theorems, complete proofs, worked examples, and extended discussions of the literature.

A note on notation. We write $I(X;Y)$ for mutual information, $H(X)$ for Shannon entropy, $D_{KL}(P \,\|\, Q)$ for Kullback–Leibler divergence, and $K(x)$ for Kolmogorov complexity. All logarithms are natural unless context specifies bits ($\log_2$). Random variables are uppercase (X, Y, T), their realisations lowercase (x, y, t), and distributions p or q. We use $0 \log 0 = 0$ throughout.

The central message of this volume is structural: information theory is not one lens among many for understanding deep learning — it is the foundational language in which the theory of deep learning will ultimately be written. Compression and prediction are dual aspects of a single optimization problem whose solution is the Gibbs algorithm.

Contents

1 The Information Bottleneck and Its Controversy 1

1.1 From Sufficient Statistics to Optimal Representations 1
1.2 The IB Functional 5
1.3 The IB Self-Consistent Equations 8
1.4 The IB Curve and Its Geometry 11
1.5 The Deep Learning Hypothesis 14
1.6 The Controversy: Three Objections 17
1.7 What Survives: IB as a Theoretical Framework 21
1.8 Extensions of the IB Framework 26
1.9 Chapter Summary 30

2 PAC-Bayes Bounds and the Prior on the Model 32

2.1 The Generalisation Gap and Why Classical Bounds Fail 32
2.2 The PAC-Bayes Framework and Main Bound 35
2.3 Gaussian Priors and Regularisation 39
2.4 The Catoni Bound and Gibbs Optimal Posterior 43
2.5 PAC-Bayes and the Loss Landscape 47
2.6 Data-Dependent Priors 52
2.7 PAC-Bayes–Information Bottleneck Connection 55
2.8 NTK Regime 58
2.9 Specialised Bounds for Deep Networks 61
2.10 Chapter Summary 64

3 Minimum Description Length and Compression 66

3.1 The Algorithmic View of Learning 66
3.2 Kolmogorov Complexity and the Universal Prior 68
3.3 Two-Part MDL 72
3.4 Stochastic Complexity and NML 75
3.5 Parametric Complexity and Fisher Information 79
3.6 MDL in Deep Learning 82
3.7 ELBO as MDL Objective 87
3.8 MDL and Occam's Razor 91
3.9 Chapter Summary 94

4 Mutual Information as a Diagnostic Tool 96

4.1 Why Mutual Information? 96
4.2 The MINE Estimator 99
4.3 KSG Estimator 103
4.4 Noise Injection 106

4.5 MI Diagnostics: Saturation, Dead Neurons, Collapse 109
4.6 The Information Plane Revisited 115
4.7 Practical MI Toolkit: CKA, V-Usable Information 118
4.8 Diagnostics for Transformers 123
4.9 Chapter Summary 127
5 Geometric Measure Theory and Representation Learning 129
5.1 Hausdorff Dimension and Effective Dimension 129
5.2 Rectifiable Sets and Network Layer Maps 133
5.3 Coarea Formula and Information Decomposition 136
5.4 Optimal Transport and Representation Comparison 139
5.5 Intrinsic Dimension Estimation 143
5.6 Covering Numbers and Dimension-Dependent Generalisation 146
5.7 Chapter Summary 149
6 Rate-Distortion Theory and Generative Models 151
6.1 The Rate-Distortion Problem 151
6.2 The Blahut-Arimoto Algorithm 154
6.3 VAEs and the Rate-Distortion Curve 157
6.4 Gaussian Water-Filling 161
6.5 GANs and Wasserstein Rate-Distortion 165
6.6 Diffusion Models as Multi-Scale Coders 169
6.7 Perception-Distortion Trade-off 173
6.8 Chapter Summary 176
7 Information-Theoretic Lower Bounds on Sample Complexity 178
7.1 Fano's Inequality 178
7.2 Le Cam's Two-Point Method 182
7.3 Assouad's Lemma 185
7.4 Deep Networks and Compositional Functions 189
7.5 MI Generalisation Bounds and IZS Method 194
7.6 Lower Bounds for Transfer Learning 198
7.7 Chapter Summary 201
8 Synthesis — A Unified Framework 203
8.1 The Four Pillars 203
8.2 The Fundamental Triangle 206
8.3 Compression-Prediction Duality 209
8.4 The Gibbs Algorithm 212
8.5 The Jeffreys Prior 215
8.6 Ten Key Theorems 218

8.7 Open Problems 222
8.8 Closing Remarks 225
9 Information Theory and Differential Privacy 227
9.1 Differential Privacy: Definitions 227
9.2 Rényi Differential Privacy 231
9.3 DP-SGD 235
9.4 Privacy-Utility as Rate-Distortion 239
9.5 Lower Bounds on Private Learning 243
9.6 Chapter Summary 246
10 Information Geometry of Neural Network Training 248
10.1 The Fisher Metric and Statistical Manifold 248
10.2 Adam as Diagonal Natural Gradient 252
10.3 Geodesics, Duality, and the EM Algorithm 256
10.4 Loss Landscape Curvature and Generalisation 260
10.5 Chapter Summary 264
11 Federated Learning and Communication-Constrained Inference 266
11.1 The Federated Setting 266
11.2 Gradient Compression as Rate-Distortion 270
11.3 Federated PAC-Bayes 274
11.4 Private Federated Learning 278
11.5 Communication Lower Bounds 282
11.6 Chapter Summary 285

Chapter 1

The Information Bottleneck and Its Controversy

Few theoretical frameworks in modern machine learning have sparked as much debate as the Information Bottleneck (IB) principle. Originally proposed by Tishby, Pereira, and Bialek in 1999 as a method for finding optimal, compact representations of data, the IB was catapulted to renewed prominence in 2017 when Tishby and Schwartz-Ziv proposed that it provides a mechanistic explanation for why deep neural networks generalise. The subsequent controversy — involving dozens of papers, competing experimental results, and fundamental mathematical objections — illuminates not just the specific question of IB in deep learning, but the broader challenge of connecting information-theoretic principles to the empirical behaviour of modern learning systems.

This chapter develops the IB framework in full mathematical detail, examines the Tishby-Schwartz-Ziv hypothesis and its critique, identifies what survives of the IB perspective, and extends the theory to continuous, multivariate, and contrastive settings. The chapter is self-contained and assumes familiarity with Shannon information theory at the level of Cover and Thomas (2006).

1.1 From Sufficient Statistics to Optimal Representations

The question of what constitutes an optimal representation of data is as old as statistics itself. Fisher (1922) posed the question in the context of parametric inference: given a family of distributions $\{p(x|\theta)\}$ indexed by a parameter θ, which functions $T(x)$ of the data preserve all the information about θ? A statistic T is sufficient for θ if the conditional distribution of X given $T(X)$ does not depend on θ:

$$p(x \mid T(x), \theta) = p(x \mid T(x)) \quad \text{for all } x, \theta.$$

Equivalently, knowing $T(x)$ is as good as knowing x for the purpose of estimating θ: the likelihood function $p(x|\theta)$ factors as $p(x|T(x)) \cdot p(T(x)|\theta)$, so $T(x)$ carries all the θ-information. The Fisher–Neyman factorisation theorem makes this operational: $T(x)$ is sufficient if and only if the likelihood can be written as

$$p(x|\theta) = g(T(x), \theta) \cdot h(x),$$

where g depends on x only through $T(x)$ and h does not depend on θ. This factorisation is both elegant and practically important: it tells us that sufficiency is equivalent to the existence of a particular algebraic structure in the likelihood.

The data processing inequality (DPI) — a cornerstone of Shannon information theory — provides the information-theoretic characterisation of sufficiency. For any Markov chain $X \to T \to Y$, the DPI states:

$$I(X; Y) \geq I(T; Y),$$

with equality if and only if $X \to T \to Y$ forms a sufficient statistic of X for Y. Here equality means that T contains all the information that X contains about Y. The DPI is

the information-theoretic analogue of the classical statement that no function of the data can contain more information about the parameter than the data itself.

1.1.1 Minimal Sufficient Statistics

Among all sufficient statistics, the minimal sufficient statistic is the one that achieves maximum compression: it discards all information in X that is irrelevant for θ, retaining only what is needed. Formally, T is a minimal sufficient statistic if it is a function of every other sufficient statistic.

The construction of minimal sufficient statistics relies on the likelihood ratio: T(x) is minimal sufficient if and only if the likelihood ratio $p(x|\theta)/p(x'|\theta)$ is independent of θ whenever T(x) = T(x'). Two data points that yield the same likelihood ratio across all θ values are informationally equivalent with respect to θ, and a minimal sufficient statistic maps them to the same value.

Example 1.1 (Exponential Family Sufficient Statistics).
For the exponential family $p(x|\eta) = h(x)\exp(\eta^\top T(x) - A(\eta))$, the statistic $T(x)$ is minimal sufficient for the natural parameter η. For Bernoulli(θ) data $X_1, \ldots, X_n$, the sufficient statistic is $T = \sum_i X_i$ (the total number of successes). For Normal(μ, σ^2) with known σ^2, the sufficient statistic is $T = \bar{X} = (1/n)\sum_i X_i$ (the sample mean). For Normal(μ, σ^2) with both unknown, the sufficient statistics are $(\bar{X}, S^2)$ where $S^2 = (1/(n-1))\sum_i (X_i - \bar{X})^2$.
In each case, T achieves maximum compression of the data while preserving all information about the parameter of interest.

The concept of sufficiency is intrinsically tied to a specific parameter θ. In practice, we rarely know the true model family, and we often care about multiple aspects of the data simultaneously. The Information Bottleneck method (Tishby, Pereira, and Bialek, 1999) generalises sufficiency to the setting where the 'parameter of interest' is a related random variable Y rather than a parametric index θ, and where we explicitly trade off compression against relevance.

1.2 The Information Bottleneck Functional

Let (X, Y) be jointly distributed with joint distribution p(x, y) over input space X and target space Y. The IB seeks a compressed representation T of X that retains maximum information about Y. The key insight is that this trade-off — between compression and relevance — can be parametrised by a single Lagrange multiplier $\beta > 0$, leading to the IB functional.

Definition 1.1 (The Information Bottleneck Functional).
Let $(X, Y) \sim p(x, y)$ and let T be a random variable induced by a Markov chain $Y \leftrightarrow X \rightarrow T$ (i.e., T is conditionally independent of Y given X). The IB functional at trade-off parameter $\beta > 0$ is
$$\mathcal{F}_\beta[p(t|x)] := I(X;T) - \beta \cdot I(T;Y).$$
The IB problem at parameter β is to find the encoder $p^*(t|x)$ that minimises $\mathcal{F}_\beta$:
$$p^*(t|x) = \operatorname{argmin}_{p(t|x)} \, [I(X;T) - \beta \cdot I(T;Y)].$$

The IB curve C_{IB} is the set of all achievable $(I(X;T), I(T;Y))$ pairs as $p(t|x)$ ranges over all stochastic maps and β ranges over $[0, \infty)$.
$C_{IB} := \{(I(X;T), I(T;Y)) : p(t|x)$ minimises F_β for some $\beta \geq 0\}$.

The functional F_β has a natural interpretation: it is the Lagrangian of the constrained optimisation problem of minimising $I(X;T)$ subject to $I(T;Y) \geq I_0$, with β being the Lagrange multiplier. Varying β from 0 to ∞ traces the Pareto frontier of the compression-relevance trade-off, which is precisely the IB curve C_{IB}.

Several structural properties of C_{IB} follow immediately from its definition:

- Concavity. The IB curve is concave in the $(I(X;T), I(T;Y))$ plane, because it is the Pareto frontier of a concave optimisation problem. Equivalently, the function $I^*(R) = \max_{p(t|x): I(X;T)\leq R} I(T;Y)$ is concave in R.
- Boundary points. At $\beta = 0$, the optimal encoder is $T \perp X$ (T is independent of X), giving the trivial point $(0, 0)$. At $\beta \to \infty$, the optimal encoder is deterministic and equals the minimal sufficient statistic of X for Y, giving the point $(H(X|Y), I(X;Y))$ (or $(H(X), I(X;Y))$ if Y is not a function of X).
- Monotonicity. As β increases from 0 to ∞, both $I(X;T)$ and $I(T;Y)$ are non-decreasing along the optimal path. More compression of X requires sacrificing information about Y.
- Achievability. Every point on C_{IB} is achievable by some stochastic encoder $p(t|x)$. The frontier is tight: there exist encoders that achieve any desired balance between compression and relevance.

The IB functional can be rewritten in a KL-divergence form that reveals its connection to variational inference. Introducing the marginal $p(t) = \int p(t|x)p(x)\,dx$ and the decoder $p(y|t) = \int p(y|x)p(x|t)p(x)\,dx \,/\, p(t)$, we have:

$$I(X;T) = E_{p(x)}[D_{KL}(p(t|x) \,\|\, p(t))],$$

$$I(T;Y) = E_{p(t)}[D_{KL}(p(y|t) \,\|\, p(y))] = E_{p(x,y)}[\log p(y|t)] - E_{p(y)}[\log p(y)].$$

Substituting, the IB functional becomes:

$$F_\beta = E_{p(x)}[D_{KL}(p(t|x) \,\|\, p(t))] - \beta\, E_{p(x,y)}[\log p(y|t)] + \beta\, H(Y),$$

which, dropping the constant $\beta H(Y)$, is equivalent to:

$$F_\beta \propto E_{p(x)}[D_{KL}(p(t|x) \,\|\, p(t))] - \beta\, E_{p(x,y)}[\log p(y|t)].$$

Remark 1.1 (Connection to the ELBO).
The KL form of F_β is structurally identical to the Evidence Lower Bound (ELBO) used in Variational Autoencoders. In the VAE setting, the first term $E[D_{KL}(q_\phi(z|x) \,\|\, p(z))]$ is the 'rate' (the information transmitted from input to latent code), and the second term $E[\log p_\theta(x|z)]$ is the 'distortion' (the reconstruction quality). The IB with the supervised decoder $p(y|t)$ playing the role of $p_\theta(x|z)$ gives the Deep Variational Information Bottleneck (VIB, Alemi et al. 2017). We return to this connection in Section 1.6.

1.3 The IB Self-Consistent Equations

The stationarity conditions for the IB functional yield a set of coupled fixed-point equations — the IB equations — that characterise the optimal encoder $p^*(t|x)$. These equations are the IB analogue of the Blahut-Arimoto equations for the rate-distortion function (Chapter 6).

Theorem 1.1 (IB Self-Consistent Equations, Tishby-Pereira-Bialek 1999). *A distribution $p^*(t|x)$ is a stationary point of F_β if and only if it satisfies the fixed-point system:*

(E1) $p^*(t|x) = p(t) / Z(x,\beta) \cdot \exp(-\beta\, D_{KL}(p(y|x) \,\|\, p(y|t)))$,

(E2) $p(t) = \int p^*(t|x)\, p(x)\, dx$,

(E3) $p(y|t) = \int p(y|x)\, p^*(x|t)\, p(x)\, dx \,/\, p(t)$,

where $Z(x,\beta) = \int p(t) \exp(-\beta\, D_{KL}(p(y|x) \,\|\, p(y|t)))\, dt$ is the normalising partition function.

Proof. We minimise $F_\beta = I(X;T) - \beta\, I(T;Y)$ subject to $\int p(t|x)\, dt = 1$ for all x. Using Lagrange multipliers $\lambda(x)$, the stationarity condition $\delta F_\beta / \delta p(t|x) = 0$ gives:

$$\log p(t|x) - \log p(t) + \beta\, D_{KL}(p(y|x) \,\|\, p(y|t)) = \lambda(x) - 1.$$

To see why, we compute the functional derivatives:

$$\delta I(X;T)/\delta p(t|x) = p(x)[\log p(t|x) - \log p(t)],$$

$$\delta I(T;Y)/\delta p(t|x) = -p(x)\, D_{KL}(p(y|x) \,\|\, p(y|t)) + \text{terms vanishing at stationarity.}$$

The second functional derivative uses the fact that $p(y|t)$ depends on $p(t|x)$ through the Bayes formula (E3), and the terms involving $\delta p(y|t)/\delta p(t|x)$ vanish when the decoder $p(y|t)$ is self-consistent. Exponentiating the stationarity condition and normalising yields (E1). □

The IB equations admit a natural iterative algorithm: initialise $p(t)$ uniformly, then alternately update (E3), (E1), (E2) until convergence. Each update step decreases F_β:

- Update $p(y|t)$: fixing $p^*(t|x)$, compute the Bayes-optimal decoder. This can only increase $I(T;Y)$ without changing $I(X;T)$, hence decreasing F_β.
- Update $p^*(t|x)$: fixing $p(t)$ and $p(y|t)$, the encoder (E1) is the unique minimiser of F_β. This can only decrease $I(X;T)$ without increasing $I(T;Y)$.
- Update $p(t)$: marginalise $p^*(t|x)$ over $p(x)$. This ensures consistency.

The algorithm is guaranteed to converge to a local minimum of F_β. For finite discrete alphabets, the global minimum is found when the algorithm is run from multiple initialisations and the solution with lowest F_β is retained.

Remark 1.2 (The partition function $Z(x,\beta)$ as a free energy). The partition function $Z(x,\beta) = \int p(t) \exp(-\beta\, D_{KL}(p(y|x) \,\|\, p(y|t)))\, dt$ has the form of a Boltzmann partition function in statistical physics, with the KL

divergence $D_{KL}(p(y|x) \,/\!/\, p(y|t))$ playing the role of an energy and β the inverse temperature. At high temperature ($\beta \to 0$), all t are equiprobable: the encoder ignores x. At low temperature ($\beta \to \infty$), the encoder is deterministic: it selects the t that minimises the KL energy for each x. The IB problem is thus formally equivalent to finding the Gibbs state of a statistical mechanical system with interaction energy given by the KL divergence between conditional distributions.

1.4 The IB Curve and Its Geometry

The IB curve C_{IB} is the central geometric object of the IB framework. Understanding its shape, boundary behaviour, and phase structure is essential for applying the IB to deep learning.

1.4.1 Endpoints and the Trivial Solutions

At $\beta = 0$, the IB functional reduces to $-I(X;T)$, which is minimised by making T independent of X. The optimal encoder is $p^*(t|x) = p(t)$ for all x, giving $I(X;T) = 0$ and $I(T;Y) = 0$. The point (0,0) is the 'complete compression' endpoint: the representation T carries no information about either X or Y.

As $\beta \to \infty$, the IB functional penalises any loss of information about Y infinitely. The optimal encoder converges to the deterministic map that minimises $I(X;T)$ subject to $I(T;Y) = I(X;Y)$. If a sufficient statistic of X for Y exists in the model class, $p^*(t|x)$ converges to this statistic, giving the point $(H(X|Y^\circ), I(X;Y))$ where Y° is the minimal sufficient statistic. If no exact sufficient statistic exists, $p^*(t|x)$ converges to the identity map $X \to T$, giving the point $(H(X), I(X;Y))$.

1.4.2 Phase Transitions Along the IB Curve

For finite discrete alphabets, the IB curve is not smooth: it exhibits kinks at critical values $\beta_1 < \beta_2 < \ldots < \beta_K$, where the optimal encoder undergoes phase transitions.

Theorem 1.2 (IB Phase Transitions, Chechik et al. 2005).
For a finite discrete source (X, Y) with joint distribution p(x, y), the optimal encoder support $|supp(p^*(t|x))| = |T^*|$ is a non-decreasing step function of β:
- *For $\beta < \beta_1$: $|T^*| = 1$ (complete compression, single codeword).*
- *For $\beta_k < \beta < \beta_{k+1}$: $|T^*| = k+1$ ($k+1$ codewords).*
- *For $\beta > \beta_K$: $|T^*| = |X|$ (no compression).*

At each critical value β_k, the IB functional F_β is non-differentiable in β (a kink in the free energy), and the system undergoes a second-order phase transition: a new 'bottleneck code' bifurcates from the existing optimal codebook.

The critical values β_k are determined by the eigenvalue structure of the stochastic matrix $p(y|x)$. Specifically, β_1 is the reciprocal of the second-largest eigenvalue of the matrix $M = \mathrm{diag}(p(x))^{1/2}\, p(x,y)\, \mathrm{diag}(p(y))^{-1}\, \mathrm{diag}(p(x))^{-1/2}$. The subsequent

critical values involve higher eigenvalues and are related to the spectral decomposition of $p(y|x)$.

The phase transition structure has a natural hierarchical interpretation. At $\beta = 0$, all inputs are merged into a single representation — maximum compression, no discrimination. As β crosses β_1, the algorithm discovers the most informative binary partition of the input space: some inputs are assigned to codeword 0 and others to codeword 1, where the partition maximises the mutual information $I(T;Y)$. As β increases further, finer partitions become optimal, revealing progressively more structure in the X-Y relationship.

This hierarchical unfolding is suggestive of the hierarchical representations learned by deep networks, where each layer discovers progressively finer features. The IB phase transitions provide a formal language for discussing this hierarchy, even if (as we discuss in Section 1.7) the direct IB interpretation of intermediate layers is problematic.

Remark 1.3 (The IB curve as a concave function).
The function $I^*(R) = \max_{p(t|x): I(X;T)\leq R} I(T;Y)$ is concave and non-decreasing in $R \in [0, H(X)]$. Concavity follows from the fact that the set of achievable $(I(X;T), I(T;Y))$ pairs is convex (as a mixture of encoders is also a valid encoder). Non-decreasingness is obvious: more compression budget allows more relevant information to be retained.
The second derivative $I^{*''}(R) \leq 0$ quantifies the marginal value of additional compression budget: as R increases, each additional bit of $I(X;T)$ buys progressively less $I(T;Y)$. At the critical values β_k, $I^{*''}(R)$ jumps from 0 to a negative value: the curve becomes strictly concave, indicating that a new code symbol has become useful.

1.5 The Deep Learning Hypothesis

In 2017, Tishby and Schwartz-Ziv proposed a striking interpretation of deep network training through the IB lens. Their hypothesis, based on empirical observations of mutual information dynamics during training, claimed that two-phase dynamics in the information plane — a fitting phase followed by a compression phase — provide the mechanistic explanation for why deep networks generalise. This section presents the hypothesis in detail and examines its empirical basis.

1.5.1 The Information Plane

For a deep network with L hidden layers $T^1, T^2, \ldots, T^L$, the information plane is the two-dimensional space with coordinates $(I(X; T^\ell), I(T^\ell; Y))$. The information plane represents each layer as a point, and the trajectory of these points during training traces out a curve in this plane.

The data processing inequality constrains the possible positions: since $Y \leftrightarrow X \to T^\ell \to T^{\ell+1} \to \hat{Y}$ forms a Markov chain, we have:

$$I(X; Y) \geq I(T^\ell; Y) \geq I(T^{\ell+1}; Y) \quad \textit{(relevance decreases with depth)},$$

$$I(X;T^\ell) \geq I(X;T^{\{\ell+1\}}) \quad \textit{(input information decreases with depth).}$$

The layers therefore appear in the information plane in a specific order: later layers have lower values of both I(X; T^ℓ) and I(T^ℓ; Y). The IB curve provides the Pareto frontier that bounds the achievable (I(X;T^ℓ), I(T^ℓ;Y)) pairs.

Hypothesis 1.1 (IB Interpretation of Deep Learning, Tishby-Schwartz-Ziv 2017).

During training of a deep feedforward network with stochastic gradient descent:
Phase 1 (Fitting, ~100–1000 epochs): I(T^ℓ; Y) increases rapidly for all layers. I(X; T^ℓ) also increases as the network learns to extract features relevant for prediction. Both fitting and compression happen simultaneously in this phase, but fitting dominates.
Phase 2 (Compression, >1000 epochs): I(T^ℓ; Y) remains approximately constant (the network has learned all it can from the training set) while I(X; T^ℓ) decreases, particularly for deeper layers. The network 'forgets' information about X that is not predictive of Y.
The compression phase was claimed to be the mechanism of generalisation: by discarding irrelevant information about X, the network reduces its variance and converges to a near-minimal sufficient statistic for the task Y.

The empirical support for this hypothesis came from experiments on a small fully-connected network (7 layers, tanh activations, 12 neurons per layer) trained on a synthetic dataset with 4096 input patterns and a binary target. Tishby and Schwartz-Ziv estimated mutual information by binning activations into 30 equally spaced bins and observed the two-phase dynamics in the information plane.

The hypothesis was immediately influential for several reasons. First, it offered a theoretical explanation for an empirical mystery: why do overparameterised networks generalise? The answer, according to the hypothesis, is that SGD implicitly implements the IB objective, driving the network toward a minimal sufficient representation that generalises well. Second, it connected deep learning to established information theory, lending theoretical credibility to empirical observations. Third, it made testable predictions about the dynamics of training, opening new avenues for empirical investigation.

1.6 The Controversy: Three Objections

The hypothesis of Tishby and Schwartz-Ziv was rapidly challenged by a series of papers that identified fundamental problems with both the empirical claims and the theoretical arguments. The controversy can be organised around three distinct objections, each of which reveals a different limitation of the IB interpretation of deep learning.

1.6.1 Objection 1: Estimator Dependence

The most immediate challenge came from Saxe et al. (2018), who showed that the two-phase dynamics are an artefact of the binning estimator used by Tishby and Schwartz-Ziv, not a genuine information-theoretic phenomenon.

Theorem 1.3 (No Compression with Non-Saturating Activations, Saxe et al. 2018).
Let T^ℓ = ReLU(W_ℓ T^{ℓ-1}) be a layer with ReLU activations. If the binned mutual information estimator is replaced by a continuous estimator (e.g., KSG or MINE), then I(X; T^ℓ) is monotone non-decreasing throughout training for generic data and weight initialisations. Conversely, for tanh activations and binning estimators, the compression artefact appears because saturating neurons (|pre-activation| >> 1) have outputs concentrated near ±1, which fall into the boundary bins of the 30-bin estimator, artificially reducing the estimated entropy H(T^ℓ).

To understand why, consider a neuron j with tanh activation and a large positive pre-activation z_j. The neuron output is tanh(z_j) ≈ 1 − 2e^{-2z_j}, which is exponentially close to 1. If we bin [−1, 1] into 30 equal bins of width 2/30 ≈ 0.067, then the neuron output falls into the top bin with probability approaching 1. The estimated entropy of this neuron (in bits) approaches $\log_2(1) = 0$. Yet the actual differential entropy of tanh(z_j) is finite and well-defined, because tanh(z_j) still varies continuously in z_j.

The binning estimator confuses saturation (a regime where the function changes slowly) with information loss (a genuine reduction in mutual information). These are fundamentally different phenomena: saturation reduces the dynamic range of the representation, but it does not necessarily reduce the information about X, because even exponentially small variations in the output are in principle recoverable.

Saxe et al. (2018) ran controlled experiments comparing tanh and ReLU activations with binning and KSG estimators. Their findings:

- Tanh + binning: two-phase dynamics observed (replicating Tishby-Schwartz-Ziv).
- Tanh + KSG: no compression phase; I(X;T) is approximately constant or slightly increasing.
- ReLU + binning: no compression phase (ReLU does not saturate).
- ReLU + KSG: no compression phase.

The conclusion is unambiguous: the compression phase is a property of the estimator and activation function combination, not a universal feature of deep network training.

1.6.2 Objection 2: Infinite Mutual Information for Deterministic Maps

A deeper theoretical objection was raised by Goldfeld and Polyanskiy (2020). Even setting aside the estimation problem, the quantity I(X; T^ℓ) is not well-defined for deterministic maps when X is continuous.

Proposition 1.1 (Deterministic Maps and Infinite MI).
Let T = f(X) be a deterministic function of a continuous random variable X with density p(x). Then:
• If f is injective (information-preserving): I(X; T) = H(X) = +∞ (the differential entropy of a continuous X is generally infinite, or at best, the MI is infinity because T determines X uniquely).

• *If f is non-injective (many-to-one): $I(X;T) = H(T) = H(f(X))$, which is finite only if $f(X)$ takes values in a finite or countable set.*
In either case, $I(X;T)$ is not a useful finite quantity for comparing the information content of different layers of a deterministic network.

Most deep network architectures, when viewed as deterministic maps from input to activation, are injective on a generic data point (because the weight matrices are generically full-rank and the activation functions are generically differentiable and non-constant). For injective maps, I(X; T) = H(X) is constant throughout training — there is no 'compression' in any meaningful sense.

Goldfeld and Polyanskiy (2020) propose resolving this by working with the noise-regularised mutual information I_σ(X; T), where a small amount of Gaussian noise is added to the activations (see Definition 4.2 in Chapter 4). This makes T^ℓ stochastic and I_σ(X; T^ℓ) well-defined and finite. However, I_σ depends on the noise level σ, which introduces an additional degree of freedom not present in the original IB framework.

1.6.3 Objection 3: SGD Does Not Optimise the IB Functional

Even if the MI were well-defined and correctly estimated, there remains the fundamental objection that SGD minimises the cross-entropy loss, not the IB functional. The cross-entropy loss is:

$$L_{CE}(w) = -E_{p(x,y)}[\log p_w(y|x)].$$

The IB functional at temperature β is:

$$F_\beta[p(t|x)] = I(X;T) - \beta \cdot I(T;Y).$$

The gradient of L_CE with respect to the network parameters w contains no term corresponding to $\partial I(X;T^\ell)/\partial w$ for any intermediate layer ℓ. There is no mechanism by which minimising L_CE would cause I(X; T^ℓ) to decrease for intermediate layers. The cross-entropy loss only cares about the mapping from X to the output $\hat{Y}$; it is completely agnostic about the information content of intermediate representations.

Any compression that occurs in intermediate layers during training with standard cross-entropy loss must be attributed to one of the following causes:

1. Implicit regularisation: SGD with finite step size and minibatch noise has a bias toward solutions that generalise well, and this bias may (coincidentally) produce representations with lower I(X; T^ℓ). This is a very different claim from the IB hypothesis.
2. Weight decay: explicit ℓ_2 regularisation penalises large weights and may reduce the effective information capacity of intermediate layers. But this is not specific to the IB objective.
3. Estimation artefacts: as discussed in Objection 1, the apparent compression is often a property of the estimator.

The IB hypothesis would be vindicated if one could show that the information-minimising trajectory traced by SGD converges to the IB curve C_IB. To date, no such result exists. The empirical evidence is that SGD trajectories in the information plane are approximately parallel to the axes — I(T^ℓ; Y) increases while I(X; T^ℓ) remains roughly constant — not that they converge to C_IB.

1.7 What Survives: IB as a Theoretical Framework

Despite the empirical controversy, the IB framework retains substantial theoretical value as a normative benchmark and a source of computationally tractable regularisers. This section identifies three surviving contributions of the IB framework to deep learning theory.

1.7.1 IB as a Lower Bound on Generalisation Error

Regardless of whether SGD implements the IB, the IB curve provides a fundamental lower bound on the achievable compression-prediction trade-off for any encoder, including neural networks. This bound is distribution-specific and can be tight.

Theorem 1.4 (IB Lower Bound on Prediction Error).
Let T be any representation of X (obtained by any encoder, deterministic or stochastic). Let $\hat{p}(y|x) = p(y|T)$ be the Bayes-optimal predictor under the representation. Then:
$E_{(x,y)\sim p}[-\log \hat{p}(y|x)] \geq H(Y|T) \geq H(Y) - I(T;Y)$.
For any fixed compression rate $R = I(X;T)$, the minimum achievable prediction error is $H(Y) - I^*(R)$, where $I^*(R)$ is the IB function evaluated at rate R.
Consequently, if a network achieves compression $I(X;T^L) = R$ in its last layer, the best possible generalisation error (in cross-entropy) is $H(Y) - I^*(R)$, regardless of architecture or training algorithm.

This theorem is tight: it is achieved by the IB-optimal encoder p*(t|x) at each rate R. For practical networks that do not implement the IB-optimal encoder, the achieved prediction error exceeds H(Y) − I*(R), representing the gap between the network's representation and the information-theoretically optimal one.

The bound is particularly useful for understanding the ceiling on task performance: if I*(R) is close to I(X; Y) for moderate R, then even a small representation can achieve near-optimal prediction. If I*(R) grows slowly with R (a 'hard' problem), then aggressive compression will always sacrifice prediction accuracy.

1.7.2 The Deep Variational Information Bottleneck

The IB objective $F_\beta = I(X;T) - \beta\, I(T;Y)$ cannot be computed exactly for high-dimensional continuous data, because mutual information between high-dimensional continuous random variables is generally intractable. However, tractable variational upper bounds on I(X;T) and lower bounds on I(T;Y) can be combined to give a computable approximation.

Alemi et al. (2017) introduced the Deep Variational Information Bottleneck (VIB), which uses:

- A variational encoder $q_\phi(t|x)$ parametrised as a Gaussian: $q_\phi(t|x) = N(\mu_\phi(x), \Sigma_\phi(x))$.
- A fixed prior $r(t)$ (typically the standard Gaussian $N(0, I)$).
- A variational decoder $p_\theta(y|t)$ parametrised as a neural network.

The VIB objective is:

$$L_{VIB}(\phi, \theta) = E_{p(x,y)}\, E_{q_\phi(t|x)}[-\log p_\theta(y|t)] + \beta \cdot E_{p(x)}[D_{KL}(q_\phi(t|x) \,/\!/\, r(t))].$$

This is a valid upper bound on F_β because:

$$I(X; T) \le E_{p(x)}[D_{KL}(q_\phi(t|x) \,/\!/\, r(t))] \quad \text{(using } r(t) \text{ as a variational marginal)},$$

$$I(T; Y) \ge E_{p(x,y)}\, E_{q_\phi(t|x)}[\log p_\theta(y|t)] - H(Y) \quad \text{(VIB lower bound on } I(T;Y)).$$

Proposition 1.2 (VIB as a Regularised Cross-Entropy Loss).
The VIB objective can be written as:
$L_{VIB} = L_{CE}(\phi, \theta) + \beta \cdot KL_{reg}(\phi)$,
where $L_{CE} = E_{p(x,y)}\, E_{q_\phi(t|x)}[-\log p_\theta(y|t)]$ *is the stochastic cross-entropy loss and* $KL_{reg} = E_{p(x)}[D_{KL}(q_\phi(t|x) \,/\!/\, r(t))]$ *is a KL regularisation term that penalises representations that deviate from the prior* $r(t)$.
Minimising L_{VIB} *with respect to* (ϕ, θ) *is therefore equivalent to minimising a regularised cross-entropy loss, where the regulariser promotes representations that are 'close' to the prior in the KL sense.*

The VIB regulariser has a geometric interpretation: $D_{KL}(q_\phi(t|x) \,/\!/\, r(t))$ measures how far the representation distribution $q_\phi(t|x)$ is from the prior $r(t) = N(0, I)$. Large KL means the representation is highly input-dependent and carries a lot of information about x. Small KL means the representation is close to the prior, compressing x aggressively.

Empirically, VIB-trained classifiers exhibit improved robustness properties compared to cross-entropy-trained classifiers:

- Robustness to adversarial perturbations: the KL regulariser prevents the encoder from producing highly-concentrated representations that are sensitive to small input changes.
- Better calibration: the stochastic encoder $q_\phi(t|x)$ produces representations with quantified uncertainty, leading to better-calibrated predictive distributions.
- Improved out-of-distribution detection: inputs that are far from the training distribution produce representations that are close to the prior (low KL), which can be used as an anomaly score.

1.7.3 IB and Adversarial Robustness

A direct theoretical consequence of the IB framework is a bound on adversarial robustness. If an encoder T compresses X to rate R (i.e., I(X;T) = R), then the encoder is insensitive to perturbations of X that lie in the 'null space' of the compression. An adversarial perturbation δ that changes T^ℓ by a large amount must also change I(X;T^ℓ) by a large amount, which is bounded by R.

Proposition 1.3 (IB Bound on Adversarial Sensitivity).
Let T be a stochastic encoder with $I(X;T) = R$ and let δ be an adversarial perturbation. The total variation distance between the representation distributions of X and $X + \delta$ satisfies:
$TV(p(T|X=x), p(T|X=x+\delta)) \leq \sqrt{(2R)} \cdot \|\delta\| / \sigma,$
where σ is the noise level of the stochastic encoder (Gaussian encoder with variance $\sigma^2 I$).
Consequently, representations with lower mutual information R are more robust to adversarial perturbations in total variation distance.

This bound formalises the intuition that compressed representations are more robust: by discarding information about X that is not predictive of Y, the encoder makes it harder for an adversary to find perturbations that change the representation without changing the label. The VIB regulariser directly implements this robustness-compression trade-off.

1.8 Extensions of the IB Framework

1.8.1 The Gaussian IB

For jointly Gaussian (X, Y) ~ N(0, Σ), the IB problem has a closed-form solution. Let Σ = [[Σ_XX, Σ_XY], [Σ_YX, Σ_YY]] be the joint covariance matrix. The conditional distribution of Y|X is also Gaussian:

$$Y|X \sim N(\Sigma_{YX} \Sigma_{XX}^{-1} X, \Sigma_{YY} - \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}).$$

Theorem 1.5 (Gaussian IB, Chechik et al. 2005).
For jointly Gaussian (X, Y), the optimal IB encoder at trade-off β is linear:
$T = A X + \text{noise}, \quad A \in \mathbb{R}^{k \times d_x},$
where A spans the top-k left singular vectors of the matrix $\Sigma_{XX}^{-1/2} \Sigma_{XY}$, ordered by singular value magnitude. The number of active directions k is determined by β: as β increases, k increases from 0 to $\text{rank}(\Sigma_{XY})$.
The IB curve for Gaussian sources coincides with the classical rate-distortion function for squared-error distortion in the directions defined by the singular vectors of $\Sigma_{XX}^{-1/2} \Sigma_{XY}$.

The Gaussian IB reduces to Canonical Correlation Analysis (CCA): the optimal linear representation of X for predicting Y is the projection onto the canonical directions of maximum correlation with Y. This provides a rigorous information-theoretic justification for CCA as a representation learning method for jointly Gaussian data.

For non-Gaussian data, the linear Gaussian IB provides a lower bound on the achievable $I(T;Y)$ at each compression rate R. The gap between the non-Gaussian IB curve and the Gaussian IB curve measures the benefit of non-linear representations: it is the mutual information that can only be extracted by non-linear encoders.

1.8.2 The Multivariate IB

When the prediction target Y is itself high-dimensional — for example, in image captioning where Y is a sentence — the scalar trade-off $\beta\, I(T;Y)$ may be too coarse. The multivariate IB extends the framework to vector-valued targets.

Given targets $Y = (Y_1, \ldots, Y_m)$, the multivariate IB solves:

$$\min_{p(t|x)} I(X;T) \quad \text{subject to} \quad I(T;Y_j) \geq R_j \quad \text{for } j = 1, \ldots, m.$$

The Lagrangian is $F_{\beta_1,\ldots,\beta_m}[p(t|x)] = I(X;T) - \sum_j \beta_j I(T;Y_j)$, and the optimal encoder satisfies a generalised version of the self-consistent equations (Theorem 1.1) with the single distortion $D_{KL}(p(y|x) \,/\!/\, p(y|t))$ replaced by a weighted sum $\sum_j \beta_j D_{KL}(p(y_j|x) \,/\!/\, p(y_j|t))$.

In deep learning, the multivariate IB is relevant for multi-task learning, where the network must simultaneously predict multiple tasks $Y_1, \ldots, Y_m$. Different tasks may require different amounts of compression of X, leading to a multidimensional trade-off surface rather than a scalar IB curve.

1.8.3 The Contrastive IB

Self-supervised contrastive learning (SimCLR, MoCo, BYOL) trains representations by maximising agreement between differently augmented views of the same image. The Contrastive IB (Wang and Isola, 2020; Tian et al., 2020) provides an information-theoretic formulation of this objective.

Let X_1 and X_2 be two augmented views of the same underlying content Z. The Contrastive IB seeks an encoder T of X_1 that:

- Maximises $I(T; X_2)$: the representation of one view should predict the other view, capturing the shared semantic content.
- Minimises $I(T; X_1)$: the representation should discard view-specific information (augmentation artefacts, noise) that is not shared between views.

Formally:

$$\min_{p(t|x_1)} I(X_1; T) \quad \text{subject to} \quad I(T; X_2) \geq I_0.$$

Proposition 1.4 (Contrastive IB and Shared Information).
The optimal Contrastive IB encoder extracts the shared mutual information $I(X_1; X_2)$ between the two views while discarding view-specific information. For augmented views where $I(X_1; X_2) = I(Z; X_1) = I(Z; X_2)$ (the views share all information about the underlying content Z), the optimal encoder satisfies:
$I(T; Z) = I(T; X_2) \approx I(Z; X_2) = I(X_1; X_2)$,
i.e., the representation captures all the task-relevant content and nothing more.

This result provides a principled explanation for why contrastive learning produces semantically meaningful representations: by maximising agreement between views, the encoder is implicitly performing IB compression with X_2 as the 'target variable'. The resulting representation captures only the view-invariant content of the image — precisely the semantic information that humans care about — while discarding low-level details.

1.9 Chapter Summary

The Information Bottleneck is one of the most influential theoretical frameworks in representation learning, combining ideas from sufficient statistics, rate-distortion theory, and variational inference into a unified optimisation principle. This chapter has developed the IB from first principles, examined its application to deep learning, and assessed the ongoing controversy surrounding that application.

The key findings are:

- The IB functional $F_\beta = I(X;T) - \beta\, I(T;Y)$ defines the compression-relevance trade-off parametrically, with the IB curve C_{IB} providing the Pareto frontier (Definition 1.1, Theorem 1.1).
- The self-consistent IB equations (Theorem 1.1) define an iterative algorithm that converges to locally optimal encoders, with the optimal solution taking the Gibbs form $p^*(t|x) \propto p(t) \exp(-\beta\, D_{KL}(p(y|x) \,/\!/\, p(y|t)))$.
- For finite alphabets, the IB curve exhibits phase transitions at critical values β_k, corresponding to the addition of new codewords (Theorem 1.2).
- The Tishby-Schwartz-Ziv hypothesis (Hypothesis 1.1) attributes the two-phase training dynamics to IB compression, but this interpretation is undermined by estimator dependence (Theorem 1.3), the ill-definedness of MI for deterministic maps (Proposition 1.1), and the absence of IB gradient signal in standard training.
- Despite the controversy, the IB provides: lower bounds on prediction error (Theorem 1.4), tractable variational regularisers (VIB, Proposition 1.2), and bounds on adversarial robustness (Proposition 1.3).
- Extensions to Gaussian sources (Theorem 1.5), multivariate targets, and contrastive learning (Proposition 1.4) demonstrate the framework's broad applicability.

The proper locus of the IB in deep learning theory is therefore not as an empirical descriptor of SGD dynamics — for which it has been largely falsified — but as a normative benchmark and a source of tractable information-theoretic regularisers. The IB tells us what an optimal representation would look like; training algorithms approximate this ideal to varying degrees.

1.2b Worked Example: Binary Classification IB

To make the IB self-consistent equations concrete, consider the simplest non-trivial case: binary classification. Let $X \in \{0,1,\ldots,7\}$ (3-bit inputs) and $Y \in \{0,1\}$ (binary label), with the joint distribution $p(x,y)$ derived from a balanced dataset. We seek a ternary representation $T \in \{A, B, C\}$ that maximally compresses X while retaining information about Y.

At $\beta = 0$, the optimal encoder is $p^*(t|x) = p(t)$ for all x — the representation T is independent of X, achieving $I(X;T) = 0$ and $I(T;Y) = 0$. This is the trivial 'ignore everything' solution.

As β increases past the first critical value $\beta_1 = 1/\lambda_1(M)$ (where λ_1 is the second-largest singular value of the normalised $p(x,y)$ matrix), the encoder bifurcates: some inputs are preferentially assigned to codeword A and others to codeword B, while codeword C remains unused. The two active codewords correspond to a binary partition of X that maximises $I(T;Y) / I(X;T)$ at slope β_1.

At $\beta_2 > \beta_1$, a second bifurcation occurs: codeword C becomes active, and the encoder now uses all three codewords. The three codewords correspond to the three natural clusters of X defined by their conditional distributions $p(y|x)$. As $\beta \to \infty$, the encoder approaches the identity map $X \to T = X$, achieving $I(X;T) = H(X)$ and $I(T;Y) = I(X;Y)$.

This worked example illustrates the key properties of the IB: hierarchical clustering (inputs with similar $p(y|x)$ are clustered together), phase transitions (the number of active codewords increases by 1 at each critical β), and the trade-off between compression and relevance (higher $\beta \to$ more codewords $\to$ more information about Y but less compression of X).

1.3b The Information Bottleneck for Continuous Variables

Extending the IB to continuous distributions requires care because the mutual information $I(X;T)$ diverges for deterministic maps. There are two main approaches.

Approach 1: Gaussian IB. For jointly Gaussian (X,Y), Theorem 1.5 gives the exact optimal encoder in closed form: it is a linear map followed by Gaussian noise. The IB curve is the classical rate-distortion function for squared-error distortion. This approach is exact but limited to Gaussian data.

Approach 2: Variational IB. For non-Gaussian data, the exact IB is intractable, and we use variational upper bounds on $I(X;T)$ and lower bounds on $I(T;Y)$. The Deep Variational Information Bottleneck (VIB, Alemi et al. 2017) parametrises the encoder as a Gaussian $q_\phi(t|x) = N(\mu_\phi(x), \Sigma_\phi(x))$ and uses the KL upper bound $I(X;T) \leq E[D_{KL}(q_\phi(t|x) \,//\, r(t))]$. The VIB objective $L_{VIB}(\phi,\theta)$ is a computable approximation to the IB functional that can be minimised by gradient descent.

Approach 3: Noise injection. Adding Gaussian noise $\varepsilon \sim N(0,\sigma^2 I)$ to the representation makes it stochastic: $T^{\ell}_\sigma = f(X) + \varepsilon$. The noise-regularised MI $I_\sigma(X;T) = H_\sigma(T) - H_\sigma(T|X)$ is finite for any $\sigma > 0$ and converges to $H(f(X))$ as $\sigma \to 0$. The noise injection approach is computationally simple but introduces a hyperparameter σ that must be chosen carefully.

The three approaches yield different approximations to the true IB curve, and they are not equivalent except in special cases. The Gaussian IB is exact for Gaussian data; VIB provides a variational upper bound on the IB functional; and noise injection provides a regularised version of the IB that depends on the noise level σ. In practice, VIB is the most widely used approach for deep learning applications.

1.4b IB and Adversarial Examples

A direct application of the IB framework is to the problem of adversarial robustness. An adversarial example is an input x + δ where ‖δ‖ is small but the network's prediction changes from the correct class to an incorrect one. The IB provides a principled framework for understanding and preventing adversarial examples.

The key insight is that adversarial examples exploit the fact that deterministic networks are injective: small changes in x produce small but non-zero changes in the representation T^ℓ, which can be amplified by subsequent layers to produce large changes in the output. A compressed representation (low I(X;T)) is inherently more robust: it maps many different inputs to the same representation, making it harder to find perturbations that change the representation significantly.

The VIB regulariser β D_KL(q_ϕ(t|x) ‖ r(t)) directly penalises representations that are highly sensitive to input changes. Large KL means the representation is highly input-dependent, which implies high sensitivity to perturbations. By minimising the VIB objective, we simultaneously improve prediction accuracy and robustness to adversarial perturbations.

Empirically, VIB-trained classifiers exhibit 20-40% better adversarial robustness (measured by accuracy under FGSM and PGD attacks) compared to cross-entropy-trained classifiers on the same architecture, confirming the theoretical prediction that compression improves robustness.

Chapter 2

PAC-Bayes Bounds and the Prior on the Model

The question of generalisation — why does a model trained on finite data perform well on unseen examples? — is the central problem of statistical learning theory. Classical answers based on uniform convergence (VC theory, Rademacher complexity) provide worst-case bounds that are vacuous for modern overparameterised networks. PAC-Bayes theory offers a radically different approach: instead of bounding the worst-case generalisation gap over all functions in a hypothesis class, it bounds the expected gap for a distribution over functions, exploiting prior knowledge about which hypotheses are likely to generalise.

This chapter develops PAC-Bayes theory from its foundations (McAllester 1999; Seeger 2002), derives the classical bounds (Theorem 2.1, Theorem 2.2), connects them to specific regularisation practices in deep learning (Sections 2.3–2.5), and establishes their relationship to the Information Bottleneck (Section 2.7). We also treat the Neural Tangent Kernel regime (Section 2.8) and data-dependent priors (Section 2.6), which are essential for obtaining non-vacuous bounds in practice.

2.1 The Generalisation Gap and Why Classical Bounds Fail

Let H be a hypothesis class and D a data distribution. A learning algorithm $A : (X \times Y)^n \to H$ maps a training set $S = \{(x_1,y_1), \ldots, (x_n,y_n)\} \in (X \times Y)^n$ to a hypothesis $h_S = A(S)$. The population risk and empirical risk of h_S are:

$$L(h_S) = \mathbb{E}_{(x,y)\sim D}[\ell(h_S(x), y)], \quad \hat{L}(h_S) = \frac{1}{n}\sum_{i=1}^{n} \ell(h_S(x_i), y_i).$$

The generalisation gap is the difference $L(h_S) - \hat{L}(h_S)$. Understanding when and why this gap is small is the fundamental problem of learning theory.

2.1.1 Classical Uniform Convergence Bounds

Classical learning theory bounds the generalisation gap by the complexity of the hypothesis class H, uniformly over all $h \in H$. The prototypical result is:

Theorem 2.1 (Rademacher Complexity Bound).
Let H be a hypothesis class with Rademacher complexity $R_n(H)$. Then for any $\delta > 0$, with probability $\geq 1 - \delta$ over $S \sim D^n$:

$$\sup_{h \in H} [L(h) - \hat{L}(h)] \leq 2R_n(H) + \sqrt{\frac{\log(2/\delta)}{2n}}.$$

The Rademacher complexity $R_n(H) = \mathbb{E}_S \mathbb{E}_\sigma[\sup_{h\in H} \frac{1}{n}|\sum_i \sigma_i h(x_i)|]$ measures how well H can fit random label noise.

For a class H parametrised by weights $w \in \mathbb{R}^p$ (e.g., a neural network), the Rademacher complexity scales as $R_n(H) = O(\sqrt{p/n})$ in the worst case — or $O(\|w\|_2/n)$ with norm constraints. For modern networks with $p = 10^8$ parameters and $n = 10^6$ training examples, this gives $R_n(H) = O(10)$ — a completely vacuous bound.

The failure of classical bounds has two root causes. First, they are worst-case over H: they must cover all possible hypotheses, including adversarially chosen ones that overfit completely. Modern trained networks are not adversarially chosen; they are the output of a specific algorithm (SGD) that has strong inductive biases. Second, they treat the network architecture and the learning algorithm as providing no information about which hypotheses are likely to be returned.

2.1.2 The PAC-Bayes Approach

PAC-Bayes theory resolves both problems by working with distributions over hypotheses rather than worst-case hypotheses. The key insight is:

- A prior P over H encodes our beliefs about which hypotheses are likely before seeing the data. If the learning algorithm returns a hypothesis close to P in KL divergence, then the hypothesis inherits the good generalisation properties of P.
- A posterior Q over H, chosen to depend on the training data S, is what the learning algorithm actually produces. The posterior concentrates on hypotheses that fit the training data well.
- The generalisation gap of the Gibbs classifier (which draws h ~ Q and returns h(x)) is bounded by the KL divergence D_KL(Q ∥ P), not the raw complexity of H.

2.2 The PAC-Bayes Framework: Definitions and the Main Bound

Definition 2.1 (Prior, Posterior, and Gibbs Classifier).
Let (H, B(H)) be a measurable hypothesis space. A prior P is a probability measure on H chosen before observing S. A posterior Q is a probability measure on H that may depend on S. The Gibbs classifier Q draws h ~ Q at each prediction and returns h(x).
The expected loss of the Gibbs classifier is:
$L(Q) := E_{h\sim Q}[L(h)] = \int_H L(h)\, dQ(h)$,
$\hat{L}(Q) := E_{h\sim Q}[\hat{L}(h)] = \int_H \hat{L}(h)\, dQ(h)$.

The PAC-Bayes framework bounds L(Q) in terms of $\hat{L}(Q)$ and D_KL(Q ∥ P). The following is the classical McAllester-Seeger bound.

Theorem 2.2 (McAllester-Seeger PAC-Bayes Bound, 1999/2002).
Let P be any prior over H chosen independently of S. Let $\ell : H \times (X\times Y) \to [0,1]$ be a bounded loss. Then for any $\delta \in (0,1)$, with probability $\geq 1-\delta$ over the draw of $S \sim D^n$, simultaneously for all posteriors Q:
$L(Q) \leq \hat{L}(Q) + \sqrt{[(D_{KL}(Q \,\|\, P) + \ln(2\sqrt{n}/\delta)) / (2n)]}$.
Equivalently, with confidence $1-\delta$:
$L(Q) - \hat{L}(Q) \leq \sqrt{[(D_{KL}(Q \,\|\, P) + \ln(2\sqrt{n}/\delta)) / (2n)]}$.

Proof sketch. The proof uses the PAC-Bayes change-of-measure inequality, which is a consequence of the Donsker-Varadhan variational formula for the KL divergence.

Step 1 (Change-of-measure lemma). For any measurable $\phi : H \to R$ and distributions P, Q over H:

$$E_{h\sim Q}[\phi(h)] \le D_{KL}(Q \,\|\, P) + \log E_{h\sim P}[e^{\phi(h)}].$$

Proof of lemma: By definition of KL divergence, $E_{h\sim Q}[\phi(h)] = E_{h\sim P}[\phi(h)(dQ/dP)(h)]$. Applying the inequality $xy \le x \log x - x + e^y$ (Young's inequality) and optimising gives the result.

Step 2. Apply the change-of-measure lemma with $\phi(h) = n(L(h) - \hat{L}(h))^2/2$ and use Hoeffding's inequality to bound:

$$E_{h\sim P}[e^{n(L(h)-\hat{L}(h))^2/2}] \le 2\sqrt{n}/\delta \quad \text{with probability} \ge 1 - \delta.$$

Step 3. Apply Jensen's inequality to convert the bound on the Gibbs classifier to a bound on the expected loss, and rearrange. □

The key point is that the bound holds uniformly over all Q — it does not depend on which posterior the algorithm chooses. This is a remarkable uniformity result: before seeing the data, we can guarantee that for any Q (however chosen after seeing S), the generalisation gap is controlled by $D_{KL}(Q \,\|\, P)$.

> **Remark 2.1 (Non-vacuousness for overparameterised networks).**
> The McAllester-Seeger bound is non-vacuous (< 1) whenever $D_{KL}(Q \,\|\, P) = o(n)$. For a Gaussian posterior $Q = N(w^*, \sigma^2 I)$ centred at the trained weights w^* and a Gaussian prior $P = N(0, \sigma_0^2 I)$,
> $D_{KL}(Q \,\|\, P) = \|w^*\|^2 / (2\sigma_0^2) + p(\sigma^2/\sigma_0^2 - 1 - \log(\sigma^2/\sigma_0^2))/2$.
> Setting $\sigma^2 = \sigma_0^2$ (posterior variance equals prior variance), this reduces to $\|w^*\|^2 / (2\sigma_0^2)$. For networks trained with weight decay, $\|w^*\|^2$ is bounded by the regularisation strength, not the parameter count p. This is the key insight: the bound is controlled by the norm of the weights, not their count.

2.3 Gaussian Priors and the Information-Theoretic Foundation of Regularisation

The PAC-Bayes bound achieves its practical utility through specific choices of prior P and posterior Q. The Gaussian family is the most analytically tractable and provides direct connections to classical regularisation methods.

2.3.1 Isotropic Gaussian Priors

Let $P = N(0, \sigma_0^2 I_p)$ be an isotropic Gaussian prior centred at the origin and $Q = N(w^*, \sigma^2 I_p)$ be an isotropic Gaussian posterior centred at the trained weights w^*. The KL divergence between two isotropic Gaussians on R^p is:

$$D_{KL}(N(w^*, \sigma^2 I) \,\|\, N(0, \sigma_0^2 I)) = (p/2)(\sigma^2/\sigma_0^2 - 1 - \log(\sigma^2/\sigma_0^2)) + \|w^*\|^2/(2\sigma_0^2).$$

This decomposition has two parts: the first term depends only on the posterior variance σ^2 relative to the prior variance σ_0^2 (it is zero when $\sigma = \sigma_0$), and the second term $\|w^*\|^2/(2\sigma_0^2)$ depends only on how far the posterior mean w^* has moved from the prior mean 0.

In the limit $\sigma = \sigma_0$ (posterior and prior have the same variance), the KL divergence reduces to $\|w^*\|^2/(2\sigma_0^2)$. This is exactly the ℓ_2 regularisation term used in weight decay! The PAC-Bayes bound with isotropic Gaussian prior thus provides the information-theoretic justification for ℓ_2 regularisation:

Proposition 2.1 (Weight Decay as PAC-Bayes Regularisation).
With prior $P = N(0, \sigma_0^2 I)$ and degenerate posterior $Q = \delta_{w^*}$ (point mass at trained weights), the PAC-Bayes bound gives:
$L(w^*) \leq \hat{L}(w^*) + \sqrt{[(\|w^*\|^2/(2\sigma_0^2) + \log(2\sqrt{n}/\delta)) / (2n)]}$.
Minimising this bound over w^* is equivalent to minimising the ℓ_2-regularised empirical risk:
$\min_{w} \hat{L}(w) + (1/(4n\sigma_0^2)) \|w\|^2$.
The optimal regularisation strength $\lambda^* = 1/(4n\sigma_0^2)$ depends on the prior variance σ_0^2 and the sample size n.

This result has several important consequences. First, it shows that the choice of regularisation strength λ has an information-theoretic interpretation: it is determined by the prior belief about the scale of the weights (σ_0^2) and the sample size n. Second, it suggests that the optimal λ decreases as n increases (more data allows larger weights), which is consistent with standard practice. Third, it shows that ℓ_2 regularisation is optimal (in the PAC-Bayes sense) for Gaussian priors, but other prior families lead to other regularisers.

2.3.2 Structured Priors and Beyond ℓ_2

The isotropic Gaussian prior treats all weight directions equally, but the inductive biases of neural networks suggest that some directions are more important than others. Structured priors can exploit this.

- Diagonal Gaussian prior $P = N(0, \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2))$: Different regularisation strengths $\lambda_i = 1/(4n\sigma_i^2)$ for different weights. This is ℓ_2 regularisation with per-parameter weights, which corresponds to structural regularisation schemes like L-BFGS with diagonal preconditioning.
- Laplace prior $P = (1/(2b)^p) \exp(-\|w\|_1/b)$: The Laplace prior corresponds to ℓ_1 regularisation (LASSO). The MAP estimator under the Laplace prior minimises $\hat{L}(w) + (1/b)\|w\|_1$, promoting sparsity.
- Spike-and-slab prior: $P = \prod_i [(1-\pi)\,\delta_0(w_i) + \pi\, N(0, \sigma_0^2)]$. This prior promotes sparse networks where a fraction $1-\pi$ of weights are exactly zero. The corresponding PAC-Bayes bound involves the number of non-zero weights rather than their squared norm.

The general principle is: the regulariser that appears in the PAC-Bayes-optimal learning objective is $-\log P(w)$ — the negative log-prior. Different choices of prior P lead to different regularisers, and the PAC-Bayes bound tells us exactly how much each choice costs in terms of generalisation.

2.4 The Catoni Bound: Tighter Bounds via Better Concentration

The McAllester-Seeger bound (Theorem 2.2) uses a square-root form that is sub-optimal in several regimes. Catoni (2007) derived a tighter bound by using a different exponential concentration inequality.

Theorem 2.3 (Catoni PAC-Bayes Bound, 2007).
For any prior P, any $\lambda > 0$, and any $\delta \in (0,1)$, with probability $\geq 1 - \delta$ over $S \sim D^n$, for all posteriors Q:
$L(Q) \leq (1/(1 - e^{-\lambda/n})) \cdot [\hat{L}(Q) + (D_{KL}(Q \| P) + \log(1/\delta)) / \lambda]$.
Optimising over λ gives a bound of the form:
$L(Q) \lesssim \hat{L}(Q) + \sqrt{[\hat{L}(Q) D_{KL}(Q \| P) + \log(1/\delta))/n]}$,
which is tighter than the McAllester-Seeger bound when $\hat{L}(Q)$ is small (near-zero training error).

The Catoni bound is adaptive: it is tighter when $\hat{L}(Q)$ is small, which is precisely the regime of modern overparameterised networks trained to near-zero training loss. The optimal λ scales as $\sqrt{(n \cdot D_{KL}/\hat{L}(Q))}$, which grows with the sample size and shrinks with the posterior's KL from the prior.

2.4.1 The Optimal Catoni Posterior: Gibbs Distribution

A remarkable property of the Catoni bound is that it identifies the optimal posterior Q* that minimises the bound for fixed P and training data S. Differentiating the bound with respect to Q under the constraint that Q is a probability measure:

Theorem 2.4 (Gibbs Posterior is Optimal).
The posterior Q^* that minimises the Catoni bound (and also the McAllester-Seeger bound) for fixed P, S, and λ is the Gibbs (Boltzmann) distribution:
$Q^*_\lambda(h) \propto P(h) \cdot \exp(-\lambda \hat{L}(h))$,
which is the Bayesian posterior under the likelihood $p(S|h) \propto \exp(-\lambda \hat{L}(h))$ and prior P.
The Catoni bound under Q^*_λ evaluates to:
$L(Q^*_\lambda) \leq -(1/\lambda) \log E_{h\sim P}[\exp(-\lambda \hat{L}(h))] + (\log(1/\delta))/\lambda$.

This theorem has a profound implication: Bayesian inference with temperature parameter λ is the PAC-Bayes-optimal learning algorithm. The Gibbs posterior is not merely a theoretical curiosity — it is the distribution that provably minimises the generalisation bound for any fixed prior P and any fixed temperature λ.

The expression $E_{h\sim P}[\exp(-\lambda \hat{L}(h))]$ is the partition function of the Gibbs distribution. Taking its negative log gives the free energy $F(\lambda) = -(1/\lambda) \log E_P[e^{-\lambda \hat{L}}]$, which bounds $L(Q^*_\lambda)$ from above. This free energy is convex in λ, and the optimal λ is found by balancing the complexity term $D_{KL}(Q^*_\lambda \| P)$ against the training loss $\hat{L}(Q^*_\lambda)$.

2.5 PAC-Bayes and the Geometry of the Loss Landscape

The connection between PAC-Bayes bounds and the geometry of the loss landscape — specifically the notion of 'flat minima' — is one of the most practically impactful results of PAC-Bayes theory. This section develops this connection in detail.

2.5.1 Flat Minima and Generalisation

The flat minima conjecture (Hochreiter and Schmidhuber, 1997) states that training algorithms that converge to flat minima of the loss landscape tend to generalise better than those that converge to sharp minima. This conjecture has been supported by extensive empirical evidence (Keskar et al., 2017) and theoretical analysis, but a rigorous justification eluded researchers for many years.

The PAC-Bayes framework provides a rigorous justification through the concept of sharpness-aware bounds.

Definition 2.2 (Sharpness at scale ϱ).
The ϱ-sharpness of the loss L at a minimum w is:*
Sharp_ϱ(w) := max_{ ‖ ε ‖ ₂ ≤ ϱ} [L(w* + ε) − L(w*)].*
A minimum is ϱ-flat if Sharp_ϱ(w) is small. A minimum is sharp if Sharp_ϱ(w*) is large even for small ϱ.*

Theorem 2.5 (Sharpness-Aware PAC-Bayes Bound, Neyshabur et al. 2017).
Let Q = N(w, ϱ²/p · I_p) and P = N(0, ϱ²/p · I_p). Assume L̂(w*) ≈ 0 (near-zero training loss). Then with probability ≥ 1 − δ over S ~ D^n:*
L(w) ≤ Sharp_ϱ(w*) + O(√[(‖ w* ‖ ²/ϱ² + log(p/δ)) / n]).*
The generalisation error is bounded by the sharpness Sharp_ϱ(w) plus a complexity term controlled by the ratio ‖ w* ‖ ²/ϱ² (the norm of the weights relative to the flatness radius).*

The proof proceeds by observing that:

4. The Gibbs classifier under Q = N(w*, ϱ²/p · I_p) achieves L(Q) ≈ L(w*) + Sharp_ϱ(w*)/2 (the expected loss under Gaussian perturbation of w* is approximately the loss at the centre plus a sharpness term).
5. D_KL(N(w*, ϱ²/p·I) ‖ N(0, ϱ²/p·I)) = p· ‖ w* ‖ ²/(2ϱ²/p) = p² ‖ w* ‖ ²/(2ϱ²) — this is large!
6. The McAllester-Seeger bound (Theorem 2.2) then gives L(w*) ≤ L(Q) + √[D_KL/(2n)], which simplifies to the stated bound.

Theorem 2.5 has a striking implication: among all minima with the same norm ‖ w* ‖ ², the one that generalises best is the flattest (largest ϱ such that Sharp_ϱ remains small). This is the PAC-Bayes justification for sharpness-aware minimisation (SAM, Foret et al. 2021).

2.5.2 SAM as PAC-Bayes Optimisation

Sharpness-Aware Minimisation (SAM) is a training algorithm that explicitly minimises the sharpness-aware loss:

$$L_SAM(w) = \max_{\|\varepsilon\|_2 \le \rho} \hat{L}(w + \varepsilon).$$

By Theorem 2.5, minimising L_SAM(w) over w is approximately equivalent to minimising the PAC-Bayes upper bound on L(w*). SAM thus has an information-theoretic interpretation: it is gradient descent on the PAC-Bayes bound, seeking a minimum that is both flat (small L_SAM) and has small norm (small $\|w^*\|^2$).

The gradient of L_SAM with respect to w is approximately:

$$\nabla_w L_SAM(w) \approx \nabla_w \hat{L}(w + \varepsilon^*(w)), \quad \varepsilon^*(w) = \rho \cdot \nabla_w \hat{L}(w) / \|\nabla_w \hat{L}(w)\|_2,$$

which perturbs the weights in the direction of maximum loss increase before computing the gradient. The two-step SAM update thus finds the steepest-ascent direction in the neighbourhood of w and then takes a gradient step to reduce the loss at that adversarial point.

2.6 Data-Dependent Priors

A fundamental limitation of the classical PAC-Bayes framework is the requirement that the prior P be chosen independently of the training data S. In practice, the most informative prior would be centred near the solution found by the learning algorithm — but this prior would depend on S.

Theorem 2.6 (Sample-Splitting Allows Data-Dependent Priors).
Let $S = S_1 \cup S_2$ be a random split of the training data with $|S_1| = m$ and $|S_2| = n - m$ ($m \le n/2$). Let $P = P(S_1)$ be any prior computed from S_1, and let $Q = Q(S_2)$ be any posterior computed from S_2. Then the McAllester-Seeger bound (Theorem 2.2) applies with n replaced by $n - m$:
$L(Q) \le \hat{L}_\{S_2\}(Q) + \sqrt{[(D_KL(Q \| P) + \log(2\sqrt{(n-m)}/\delta)) / (2(n-m))]}$.
The bound uses only $|S_2| = n - m$ samples for training and evaluation.

The sample-splitting scheme trades off m samples 'for the prior' against n − m samples for the bound. The optimal split depends on how much the data-dependent prior improves the bound. Typically, a small fraction $m = O(\sqrt{n})$ is sufficient: even a warm-up training run on $\sqrt{n}$ samples can identify the basin of attraction and define an informative prior.

Corollary 2.1 (Non-Vacuous Bounds via Optimised Posteriors).
Dziugaite and Roy (2017) demonstrated that the McAllester-Seeger bound can be made non-vacuous (strictly less than 1) for networks with $p = 10^6$ parameters on MNIST ($n = 5\times10^4$), by:
1. Splitting: use S_1 (first 5×10^3 examples) to define $P = N(w_init, \sigma_0^2 I)$.
2. Training: use S_2 to minimise $L_PAC\text{-}Bayes(Q) = \hat{L}_\{S_2\}(Q) + D_KL(Q \| P)/(2(n-m))$ directly.
3. Bound: evaluate $\hat{L}_\{S_2\}(Q) + \sqrt{[D_KL(Q \| P)/(2(n-m))]}$.

The result: a non-vacuous generalisation bound of ≈ 26% test error, compared to the actual test error of ≈ 2%. This was the first rigorous, non-trivial generalisation guarantee for overparameterised neural networks.

The gap between the bound (26%) and the actual error (2%) reflects the looseness of the PAC-Bayes bound in its current form. Subsequent work has improved this gap significantly, but a tight PAC-Bayes bound that matches the actual generalisation error of deep networks remains an open problem.

2.7 The PAC-Bayes–Information Bottleneck Connection

The connection between PAC-Bayes and IB is not merely analogical — it is algebraic. The Catoni optimal posterior $Q^*_\lambda \propto P\, e^{\{-\lambda L\}}$ is precisely the IB solution when the IB is applied to the weight space w rather than the activation space T.

Proposition 2.2 (Weight-Space IB = PAC-Bayes).
Define a 'weight-space IB' problem: the source is $w \sim P$ (drawn from the prior), the target is the training accuracy acc(w, S), and the stochastic map is $Q(w)$ (the posterior over weights). The weight-space IB functional at trade-off $\beta = \lambda$ is:
$F_\beta[Q] = D_{KL}(Q \,\|\, P) - \beta\, E_Q[acc(w, S)]$.
The Catoni optimal posterior $Q^*_\lambda \propto P\, e^{\{-\lambda L\}}$ minimises this functional at trade-off $\beta = \lambda$:
$Q^*_\lambda = \operatorname{argmin}_Q [D_{KL}(Q \,\|\, P) - \lambda\, E_Q[acc(w, S)]]$.
PAC-Bayes optimisation in weight space is therefore IB optimisation in weight space, with the prior P playing the role of the input distribution and the training accuracy playing the role of the relevance variable Y.

This connection illuminates a fundamental duality between the IB and PAC-Bayes frameworks:

- IB in activation space: studies the information trade-off within a fixed network, asking how much input information is retained in each layer representation.
- PAC-Bayes in weight space: studies the information trade-off across the ensemble of networks, asking how much training data information is retained in the posterior distribution over weights.

Both frameworks reduce to the same Gibbs distribution — one over representations $p^*(t|x)$ and the other over weights $Q^*_\lambda(w)$ — reflecting the fundamental universality of the Gibbs form as the solution to information-theoretic optimisation problems.

2.8 PAC-Bayes in the Neural Tangent Kernel Regime

The Neural Tangent Kernel (NTK) theory (Jacot et al., 2018) provides a linearisation of neural network training dynamics in the infinite-width limit. In this regime, the network function $f_w(x)$ evolves linearly in the weights according to:

$$f_{w_t}(x) = f_{w_0}(x) + \nabla_w f_{w_0}(x)^T (w_t - w_0) + O(1/\sqrt{width}),$$

and gradient descent on the squared loss converges to the minimum-norm interpolant (the solution with minimum $\|w - w_0\|_2$ that achieves zero training loss). The NTK matrix $K(x_i, x_j) = \nabla_w f_{w_0}(x_i)^T \nabla_w f_{w_0}(x_j)$ controls the convergence rate and the implicit bias of gradient descent.

Theorem 2.7 (NTK PAC-Bayes Bound).
In the NTK regime, let $P_{NTK} = N(w_0, \sigma_0^2 I)$ be the prior (initialisation distribution) and $Q_{NTK} = N(w^*, \sigma^2 I)$ be the posterior (Gaussian perturbation of the minimum-norm interpolant w^*). The PAC-Bayes bound evaluates to:
$L(w^*) \le \hat{L}(w^*) + \sqrt{[(\|w^* - w_0\|^2_{K_{NTK}^{-1}} / (2\sigma_0^2) + \log(2\sqrt{n}/\delta)) / (2n)]}$,
where $\|v\|^2_{K^{-1}} = v^T K^{-1} v$ is the NTK-norm. The generalisation bound is controlled by the NTK-norm of the weight displacement $w^* - w_0$, not the Euclidean norm $\|w^* - w_0\|_2$.

The NTK-norm $\|w^* - w_0\|_{K_{NTK}^{-1}}$ is reparametrisation-invariant: if we reparametrise the network as $\varphi = f(w)$, the NTK-norm in φ-coordinates equals the NTK-norm in w-coordinates. This is a significant improvement over the raw ℓ_2 norm $\|w^* - w_0\|_2$, which depends on the choice of parametrisation.

In the NTK regime, the minimum-norm interpolant w^* satisfies $w^* - w_0 = \tilde{K}(\tilde{K}^T \tilde{K})^{-1} y$, where $\tilde{K} \in \mathbb{R}^{p \times n}$ is the NTK feature matrix and $y \in \mathbb{R}^n$ is the label vector. The NTK-norm is:

$$\|w^* - w_0\|^2_{K_{NTK}^{-1}} = y^T (\tilde{K}\tilde{K}^T)^{-1} y = y^T K^{-1} y,$$

where $K = \tilde{K}^T \tilde{K}/p \in \mathbb{R}^{n \times n}$ is the empirical NTK matrix. This is exactly the squared rkHS norm of the predictor in the kernel space defined by the NTK. The PAC-Bayes bound is thus equivalent to the kernel ridge regression bound with the NTK as the reproducing kernel.

Remark 2.2 (Beyond the NTK regime).
The NTK regime provides a tractable but potentially loose approximation to the true training dynamics of finite-width networks. Beyond the NTK regime, feature learning occurs: the network's representations change during training, and the NTK matrix $K(x_i, x_j)$ is no longer constant. In this regime, the PAC-Bayes bound requires tracking the evolution of the posterior Q_t as training progresses, which is significantly more complex. The 'mean field' theory of neural networks (Mei et al., 2018) provides an alternative regime where the weights are exchangeable and the posterior over weights factorises, allowing for tractable PAC-Bayes analysis beyond the NTK.

2.9 Specialised Bounds for Deep Networks

Beyond the general McAllester-Seeger and Catoni bounds, a number of specialised PAC-Bayes bounds exploit the specific structure of deep networks to obtain tighter guarantees.

2.9.1 Norm-Based Bounds for Deep Networks

Bartlett et al. (2017) derived generalisation bounds for deep networks that scale with the product of spectral norms of the weight matrices, rather than the total number of parameters. For an L-layer network with weight matrices $W_1, \ldots, W_L$:

$$\textit{Generalisation gap} \leq O(\sqrt{n}) \cdot \|W_1\|_\sigma \cdot \|W_2\|_\sigma \cdot \ldots \cdot \|W_L\|_\sigma,$$

where $\|W\|_\sigma$ is the spectral norm (largest singular value) of W. This bound can be derived from a PAC-Bayes argument with a Gaussian prior whose scale σ_0 is set by the spectral norm product.

Theorem: The PAC-Bayes bound for a deep network with prior $P = N(0, (\varrho^2/p)I)$ and posterior $Q = N(w^*, (\varrho^2/p)I)$, where $\varrho = \varrho_0 / \|W_1\|_\sigma \cdots \|W_L\|_\sigma$, gives:

$$L(w^*) \leq \hat{L}(w^*) + \tilde{O}\left(\|W_1\|_\sigma \cdots \|W_L\|_\sigma \cdot \sqrt{p \log p \,/\, n}\right).$$

2.9.2 Compression-Based Bounds

Arora et al. (2018) derived a PAC-Bayes bound based on network compression: if a network can be compressed to a description of length k bits with training error ε, then:

$$L(w^*) \leq \varepsilon + \sqrt{\left(k \log 2 + \log(1/\delta)\right) / (2n)}.$$

This is an MDL-style bound (Chapter 3) derived within the PAC-Bayes framework. It is tight when the network admits a short description that captures all its predictive power — which is precisely what network quantisation and pruning aim to achieve.

2.10 Chapter Summary

PAC-Bayes theory provides the most powerful currently available framework for obtaining non-vacuous generalisation bounds for overparameterised neural networks. The key contributions of this chapter are:

- The McAllester-Seeger bound (Theorem 2.2) and Catoni bound (Theorem 2.3) bound the population risk $L(Q)$ in terms of the KL divergence $D_KL(Q \| P)$ and the empirical risk $\hat{L}(Q)$.
- The optimal posterior for the Catoni bound is the Gibbs distribution $Q^*_\lambda \propto P\, e^{-\lambda \hat{L}}$ (Theorem 2.4), unifying Bayesian inference and PAC-Bayes optimisation.
- Weight decay (ℓ_2 regularisation) is the PAC-Bayes-optimal learning objective under a Gaussian prior (Proposition 2.1), and other regularisers correspond to other prior families.
- Flat minima generalise better because sharpness controls the KL divergence of a perturbation posterior from the prior (Theorem 2.5), providing the rigorous justification for SAM.

- Data-dependent priors via sample splitting (Theorem 2.6) enable non-vacuous bounds for million-parameter networks (Corollary 2.1).
- The PAC-Bayes optimal posterior in weight space is the weight-space IB solution (Proposition 2.2), establishing a formal duality between the IB and PAC-Bayes frameworks.
- In the NTK regime, the PAC-Bayes bound is controlled by the NTK-norm of the weight displacement, providing a reparametrisation-invariant complexity measure (Theorem 2.7).

2.3b Laplace Priors and ℓ_1 Regularisation

The PAC-Bayes framework extends naturally beyond Gaussian priors to any distribution over the hypothesis space. The Laplace distribution provides the information-theoretic foundation for ℓ_1 regularisation (LASSO).

Let P = Laplace(0, b)^p = (1/(2b))^p exp(− // w // ₁/b) be the isotropic Laplace prior on R^p. The MAP estimator under this prior is θ̂_MAP = argmax_w [log P(w) + log p(S|w)] = argmin_w [−log p(S|w) + // w // ₁/b], which is exactly ℓ_1-regularised empirical risk minimisation with regularisation strength λ = 1/b.

The KL divergence between a Gaussian posterior Q = N(w*, σ²I) and the Laplace prior P = Laplace(0,b)^p is: D_KL(Q // P) = // w* // ₁/b + (p/2) log(2b²/(πeσ²)) + O(// w* // ₂/b). For sparse w* (many coordinates near zero), // w* // ₁ is much smaller than // w* // ₂, giving a tighter PAC-Bayes bound under the Laplace prior than under the Gaussian prior.

The sparsity-promoting property of the Laplace prior arises from its sharp peak at 0 and heavy tails: the prior assigns high probability to weights that are either exactly zero or very large, encouraging sparse solutions where most weights are near zero and a few are large. This is the PAC-Bayes justification for the LASSO — it is the optimal regulariser for Laplace priors.

2.5b The SAM Algorithm: Theory and Practice

Sharpness-Aware Minimisation (SAM, Foret et al. 2021) explicitly minimises the sharpness-aware loss L_SAM(w) = max_{ // ε // ≤ϱ} L̂(w+ε). The gradient of L_SAM is:

$$\nabla_w L_SAM(w) \approx \nabla_w \hat{L}(w + \varepsilon^*(w)), \quad \varepsilon^*(w) = \varrho \cdot \nabla_w \hat{L}(w) \,/\, \| \nabla_w \hat{L}(w) \|_2,$$

so SAM computes the gradient at the worst-case perturbation point w + ε*(w) rather than at w. This requires two forward-backward passes per step (one to find ε*(w), one to compute the gradient), making SAM approximately 2× more expensive per step than SGD.

The PAC-Bayes justification for SAM (Theorem 2.5) shows that minimising L_SAM(w) is approximately equivalent to minimising the PAC-Bayes bound. The key insight is that

Sharp_ϱ(w*) = L_SAM(w*) − L̂(w*) controls the generalisation bound, so minimising L_SAM directly minimises the bound.

Empirical results: on ImageNet with ResNet-50, SAM achieves 1.5–2% better top-1 accuracy than SGD, and on NLP benchmarks (GLUE), SAM fine-tuning of BERT achieves 0.5–1% better accuracy. The gains are larger for smaller datasets (where generalisation is more challenging) and smaller for very large datasets (where all algorithms approach the same Bayes-optimal accuracy).

Extensions of SAM: m-SAM applies the worst-case perturbation only to a randomly selected subset of parameters; ASAM (adaptive SAM) normalises the perturbation by the parameter scale; SAM++ uses higher-order information about the loss landscape. All of these variants can be understood as approximations to the PAC-Bayes minimisation problem under different prior-posterior configurations.

2.7b PAC-Bayes and Stochastic Training

The PAC-Bayes framework provides a principled interpretation of several stochastic training techniques that are used in practice but whose theoretical basis was unclear.

Dropout as a posterior distribution: applying dropout with rate p to a network produces a random subset of neurons at each forward pass, effectively sampling from a distribution over architectures. The dropout posterior Q_dropout assigns probability ∝ (p^k (1-p)^{d-k}) to any sub-network with k active neurons (out of d total). The PAC-Bayes bound under Q_dropout involves D_KL(Q_dropout ∥ P), where P is the prior over sub-networks. This gives a rigorous generalisation bound for dropout-trained networks.

Stochastic weight averaging (SWA, Izmailov et al. 2018): averaging multiple checkpoints of a training trajectory produces an estimator that lies at the centre of a flat region of the loss landscape. From the PAC-Bayes perspective, SWA can be understood as approximating the Gibbs posterior by taking the centroid of the samples drawn by SGD. The centroid of a flat region corresponds to a larger effective ϱ in Theorem 2.5, giving better PAC-Bayes bounds.

Label smoothing: replacing one-hot labels y with (1-α) y + α/C (where C is the number of classes) reduces the cross-entropy loss and encourages flatter minima. From the PAC-Bayes perspective, label smoothing reduces the effective empirical risk L̂(Q), which tightens the Catoni bound (Theorem 2.3) through the √L̂(Q) factor.

Chapter 3

Minimum Description Length and Compression as a Unifying Principle

The Minimum Description Length (MDL) principle is the formalisation of Occam's Razor in the language of information theory: among all hypotheses consistent with the data, prefer the one that enables the shortest joint description of the hypothesis and the data. This is not a heuristic preference for simplicity — it is a provably optimal strategy for minimax-regret prediction, and it subsumes both maximum likelihood estimation and Bayesian inference as special cases.

The intellectual genealogy of MDL is rich. Kolmogorov (1965) and Solomonoff (1964) independently developed the notion of algorithmic complexity — the length of the shortest program that outputs a given string — as the theoretically ideal notion of description length. Rissanen (1978) made this computable by replacing Kolmogorov complexity with model-class stochastic complexity, defined via the normalised maximum likelihood (NML) distribution. Barron (1985) proved that the MDL estimator achieves the minimax rate of convergence. The modern NML framework (Rissanen 1996; Barron 1998) provides a complete theory connecting MDL to the geometry of statistical models via the Fisher information.

3.1 The Algorithmic View of Learning

The starting point of MDL is the observation that describing data and describing a model are not independent activities: a good model enables a compact description of the data (the model predicts the data well, leaving little residual to describe), while a compact model requires few bits to describe itself. The total description length — model + data given model — is the quantity to minimise.

This view connects directly to Shannon's source coding theorem: the optimal code for a source X with distribution p assigns code length $-\log_2 p(x)$ to symbol x. If we know the true distribution p, we can achieve the entropy H(X) bits per symbol on average. If we use an incorrect model q, we pay an extra $D_{KL}(p \,//\, q)$ bits per symbol — the 'cost of ignorance'. MDL minimises this total cost by choosing q to minimise the two-part description length: (bits to describe q) + (bits to describe x under q).

3.2 Kolmogorov Complexity and the Universal Prior

Definition 3.1 (Kolmogorov Complexity).
Fix a universal Turing machine U. The Kolmogorov complexity of a binary string $x \in \{0,1\}^*$ is $K(x) := \min\{|\pi| : U(\pi) = x\}$, where $|\pi|$ is the bit length of program π. The conditional Kolmogorov complexity of x given y is $K(x|y) := \min\{|\pi| : U(\pi, y) = x\}$. Kolmogorov complexity is the length of the shortest program that outputs x. It satisfies $K(x) \leq |x| + O(1)$ (trivial encoding) and $K(x) = O(\log|x|)$ for highly regular strings.

Kolmogorov complexity is non-computable (by reduction to the halting problem: if we could compute K(x) we could solve the halting problem). However, it serves as the theoretically ideal target that computable description length measures seek to approximate. Several key properties:

Proposition 3.1 (Symmetry of Information).
Up to an additive constant $O(\log \max(K(x), K(y)))$:
$K(x,y) = K(x) + K(y|x) = K(y) + K(x|y)$.
This is the algorithmic analogue of the chain rule $H(X,Y) = H(X) + H(Y|X)$ and provides the foundation for algorithmic mutual information:
$I_K(x;y) := K(y) - K(y|x) = K(x) - K(x|y)$.

The algorithmic data processing inequality follows: for any computable function f, K(f(x)|y) ≤ K(x|y) + O(1). This mirrors the Shannon DPI and establishes that computable transformations cannot increase algorithmic information.

Solomonoff's universal prior M(x) = 2^{−K(x)} assigns higher probability to simpler strings. A remarkable property: the Kolmogorov complexity K(x) equals $-\log_2 M(x)$ up to a constant, so M is the 'ideal prior' that assigns highest probability to the simplest explanations. M dominates all computable probability distributions in the sense that M(x) ≥ c_Q · Q(x) for any computable distribution Q, where c_Q depends on Q but not on x.

3.3 Two-Part MDL: Hypothesis and Data Encoding

Two-part MDL instantiates the compression ideal in a computable framework. Given a model class $M = \{p(x|\theta) : \theta \in \Theta\}$ and data x^n:

- Part 1: Encode the parameter θ with $L_1(\theta)$ bits (a prefix-free code over Θ).
- Part 2: Encode the data x^n with $L_2(x^n|\theta) = -\log_2 p(x^n|\theta)$ bits (Shannon code under model θ).

The two-part MDL estimator minimises the total description length:

$$\hat{\theta}_{MDL} = \operatorname{argmin}_{\theta\in\Theta} [L_1(\theta) + L_2(x^n|\theta)] = \operatorname{argmin}_{\theta\in\Theta} [L_1(\theta) - \log p(x^n|\theta)].$$

Proposition 3.2 (MDL and MAP Equivalence).
If $L_1(\theta) = -\log p(\theta)$ for a probability measure p over Θ (i.e., the prior encodes the parameter), then $\hat{\theta}_{MDL}$ coincides with the MAP estimator $\hat{\theta}_{MAP} = \operatorname{argmax}_\theta [\log p(\theta) + \log p(x^n|\theta)]$.
Conversely, any prefix-free code L_1 defines a 'generalised prior' via Kraft's inequality $\Sigma_\theta 2^{-L_1(\theta)} \le 1$, and the MDL estimator under L_1 is the MAP estimator under this generalised prior.

Two-part MDL has a fundamental limitation: it treats the encoding of θ and x^n|θ as independent operations, but the optimal encoding of x^n depends on θ (shorter when the model fits well) and the optimal encoding of θ depends on the data (more precision

is needed when the data is informative about θ). The NML framework resolves this coupling.

3.4 Stochastic Complexity and the Normalised Maximum Likelihood

Definition 3.2 (NML Distribution and Stochastic Complexity).
The Normalised Maximum Likelihood (NML) distribution for model class $\mathcal{M}$ is:
$p_{\mathrm{NML}}(x^n) := \max_\theta p(x^n \mid \theta) \,/\, C_n(\mathcal{M}), \quad C_n(\mathcal{M}) := \sum_{z^n \in \mathcal{X}^n} \max_\theta p(z^n \mid \theta).$
The normalising constant $C_n(\mathcal{M}) = \sum_{z^n} p(z^n \mid \hat{\theta}(z^n))$ is the parametric complexity of $\mathcal{M}$.
The stochastic complexity of x^n is $\mathrm{SC}(x^n \mid \mathcal{M}) := -\log p_{\mathrm{NML}}(x^n) = -\log p(x^n \mid \hat{\theta}(x^n)) + \log C_n(\mathcal{M})$.

Theorem 3.1 (Rissanen's Minimax Theorem).
The NML code achieves the minimax regret over all sequences:
$p_{\mathrm{NML}} = \arg\min_q \max_{x^n} \left[-\log q(x^n) + \log p(x^n \mid \hat{\theta}(x^n))\right],$
and the minimax regret equals $\log C_n(\mathcal{M})$. No other code can have lower worst-case regret than $\log C_n(\mathcal{M})$.
Proof: The regret of code q on x^n is $R(q, x^n) = -\log q(x^n) - \max_\theta \log p(x^n \mid \theta)$. For the NML code, $R(p_{\mathrm{NML}}, x^n) = \log C_n(\mathcal{M})$ is constant. Any other code must have higher regret on some sequence by the minimax theorem for zero-sum games. □

The minimax regret interpretation gives MDL a game-theoretic foundation: the NML code is the optimal hedge against an adversary who chooses the worst-case data sequence after seeing the code. This is analogous to the minimax interpretation of PAC learning, but for prediction rather than classification.

3.5 The Parametric Complexity and Fisher Information

Theorem 3.2 (Asymptotic Complexity, Rissanen 1996).
For a smooth regular parametric family $\{p(x \mid \theta) : \theta \in \Theta \subseteq \mathbb{R}^d\}$ satisfying standard regularity conditions:
$\log C_n(\mathcal{M}) = \frac{d}{2} \log \frac{n}{2\pi} + \log \int_\Theta \sqrt{\det I(\theta)}\, d\theta + o(1),$
where $I(\theta) = \mathbb{E}_\theta[-\nabla^2_\theta \log p(x \mid \theta)]$ is the $d \times d$ Fisher information matrix.
The leading term $\frac{d}{2} \log n$ matches the BIC penalty; the subleading term is the log-volume of the model manifold under the Jeffreys prior $\sqrt{\det I(\theta)}\, d\theta$.

The Fisher information volume $\int_\Theta \sqrt{\det I(\theta)}\, d\theta$ is the Riemannian volume of the statistical manifold under the Fisher metric. Models that have high Fisher information (sharply peaked likelihoods, well-identifiable parameters) have large Fisher information volume and are therefore intrinsically more complex in the MDL sense — even if they have the same number of parameters as simpler models.

Corollary 3.1 (BIC as First-Order MDL).
The Bayesian Information Criterion BIC(x^n|M) := −log p(x^n|θ̂) + (d/2) log n is the first-order MDL approximation to the stochastic complexity, obtained by discarding the Jeffreys prior term.
BIC over-penalises models with low Fisher information (uninformative directions in parameter space) and under-penalises models with high Fisher information (highly informative directions). The NML stochastic complexity is the correct, information-geometry-aware model selection criterion.

3.6 MDL in Deep Learning: The Network as a Compressor

3.6.1 The Two-Part MDL View of Neural Networks

A trained neural network $f_w : X \to Y$ encodes a hypothesis about the mapping $X \to Y$. In the MDL framing, there are two description costs:

- Part 1 (Model description): Encoding w at b bits per weight costs $L_1(w) = b \cdot p$ bits. For a standard 32-bit float network with p parameters, $L_1 = 32p$ bits. Quantised networks (b = 4, 8 bits) reduce this cost.
- Part 2 (Data given model): Encoding the training targets y^n given the inputs x^n and the model w costs $L_2(y^n|x^n, w) = -\log p(y^n|x^n, w) = n \cdot \hat{L}(w) / \log 2$ bits (in bits), where $\hat{L}(w)$ is the cross-entropy training loss.

The two-part MDL objective is:

$$\min_w [b \cdot p + n \cdot \hat{L}(w) / \log 2].$$

Setting the derivative with respect to b to zero gives $b^* = \log 2 \cdot n \cdot |d\hat{L}/db| / p$ — the optimal precision balances the model complexity cost (increases with b) against the data description cost (decreases with b as higher precision allows better fitting).

In the overparameterised regime ($p \gg n$), the MDL objective is dominated by the model description cost $L_1(w) = bp$. Minimising MDL therefore pushes toward sparser, lower-precision networks that still fit the data well — exactly what pruning, quantisation, and knowledge distillation achieve.

Theorem 3.3 (MDL Generalisation Bound, Barron and Cover 1991).
Let H be a countable hypothesis class with prefix-free code lengths $\{L(h) : h \in H\}$ satisfying Kraft's inequality $\Sigma_h 2^{-L(h)} \leq 1$. Then for any $\delta \in (0,1)$, with probability $\geq 1 - \delta$ over $S \sim D^n$:
$L(h) \leq \hat{L}(h) + \sqrt{[(L_code(h) \log 2 + \log(1/\delta)) / (2n)]}$.
The bound applies simultaneously to all $h \in H$ without additional assumptions on the function class.

The MDL bound differs from PAC-Bayes in that L_code(h) is a fixed, data-independent quantity. This makes it simpler to evaluate but potentially looser: the PAC-Bayes posterior Q can be optimised over training data S, while the MDL code is fixed before

observing S. The two approaches are related by: L_code(h) = −log P(h) under the MDL prior P(h) = 2^{−L_code(h)}, and D_KL(δ_h ∥ P) = −log P(h) = L_code(h) for the point-mass posterior.

3.6.2 Prequential MDL and Online Learning

The two-part MDL requires specifying a code for w before seeing the data. Prequential (predictive) MDL avoids this by encoding data sequentially:

$$L_{\text{prequential}}(x^n) := \sum_{t=1}^{n} [-\log p(x_t \mid x_{<t}, \hat{\theta}(x_{<t}))],$$

where θ̂(x_{<t}) is the MLE computed from the first t−1 observations. This is the cumulative log-loss of the online MLE. For i.i.d. data, L_prequential(x^n) → SC(x^n|M) as n → ∞ by the asymptotic equivalence of the NML code and the prequential code.

The prequential MDL has a direct connection to online learning: the total code length L_prequential(y^n|x^n) is the cumulative loss of the online prediction algorithm that uses the MLE at each step. An algorithm with low prequential code length is an efficient online learner — it predicts each new data point using the best model fitted to previous data.

3.7 The ELBO of VAEs as an MDL Objective

The connection between MDL and variational autoencoders (VAEs) is one of the deepest results in the theory of generative models. It establishes that the VAE objective is not merely a variational approximation to the marginal likelihood — it is an MDL objective for latent variable models.

Definition 3.3 (Bits-Back Coding, Hinton and van Camp 1993).
In bits-back coding, the receiver knows the input x and can use the approximate posterior $q(z|x)$ to 'get back' $H(q(z|x))$ bits for free. The net communication cost of transmitting the latent variable z is:
$L_{\text{net}}(z|x) = -\log p(z) - (-H(q(z|x))) = D_{KL}(q(z|x) \,\|\, p(z))$.
The total description length for x under the latent variable model is:
$L_{\text{BB}}(x) = D_{KL}(q(z|x) \,\|\, p(z)) + E_{q(z|x)}[-\log p(x|z)] = -L_{\text{ELBO}}(x)$.

Theorem 3.4 (ELBO = MDL for Latent Variable Models).
The negative ELBO $-L_{\text{ELBO}}(x) = E_q[-\log p(x|z)] + D_{KL}(q(z|x) \,\|\, p(z))$ equals the two-part MDL code length for the pair (x, z) under the bits-back scheme:
$-L_{\text{ELBO}} =$ *[latent code cost: $D_{KL}(q(z|x) \,\|\, p(z))$] + [reconstruction cost: $E_q[-\log p(x|z)]$].*
Maximising the ELBO (minimising $-L_{\text{ELBO}}$) is therefore equivalent to minimising the total description length of x under the latent variable model.

This theorem has profound implications for the interpretation of VAE training:

- The KL term D_KL(q(z|x) ∥ p(z)) is the rate — the number of bits used to encode the latent representation z, above the 'baseline' of the prior p(z).

- The reconstruction term $E_q[-\log p(x|z)]$ is the distortion — the number of bits needed to describe x given the representation z.
- The β-VAE (Higgins et al., 2017) with parameter β weights these two terms differently: high β prioritises low rate (compressed representations) at the cost of high distortion (blurry reconstructions). This is precisely the rate-distortion trade-off of Chapter 6.

The MDL interpretation also clarifies the disentanglement objective of VAEs. A disentangled representation is one where different dimensions of z capture independent factors of variation in x. In the MDL framework, disentanglement corresponds to a factored prior $p(z) = \prod p(z)$ and a factored posterior $q(z|x) = \prod q(z|x)$. The KL term decomposes as $D_{KL}(q(z|x) \,/\!/\, p(z)) = \Sigma\, D_{KL}(q(z|x) \,/\!/\, p(z)) + TC(z)$, where $TC(z) = D_{KL}(q(z|x) \,/\!/\, \prod q(z|x))$ is the total correlation — a measure of statistical dependence among the dimensions of z. Disentanglement corresponds to low TC(z).

3.8 MDL, Occam's Razor, and Bias-Variance Trade-off

The MDL principle is the operational form of Occam's Razor: prefer the simplest explanation consistent with the data, where 'simplest' is measured in bits of description length. This is not a metaphysical commitment to simplicity but a consequence of Rissanen's minimax theorem: the NML code minimises the worst-case redundancy, and simpler models (lower stochastic complexity) achieve this bound with fewer bits.

Theorem 3.5 (MDL Consistency, Rissanen 1978).
Let $M_1 \subset M_2 \subset \ldots$ be a nested sequence of model classes and $p^* \in M_{k^*} \setminus M_{k^*-1}$ the true DGP. The MDL model selector $\hat{k}_{MDL} = \operatorname{argmin}_k SC(x^n|M_k)$ is consistent: $P(\hat{k}_{MDL} = k^*) \to 1$ as $n \to \infty$.
The penalty $(d_k/2)\log n$ in BIC ensures that the marginal gain in log-likelihood from adding parameters grows as $O(1)$ while the complexity penalty grows as $O(\log n)$, so BIC (and MDL) correctly identifies the true model order.

The MDL trade-off has a clean bias-variance interpretation:

- Underfitting ($k < k^*$): large L_2 (the model predicts poorly, requiring many bits to describe the residuals) and small L_1 (few parameters to describe). The bias dominates.
- Correct model ($k = k^*$): the optimal balance between L_1 and L_2. Both bias and variance are controlled.
- Overfitting ($k > k^*$): small L_2 (the model fits the data perfectly) and large L_1 (many parameters, requiring many bits to encode). The variance dominates.

The MDL perspective unifies the bias-variance trade-off with information theory: the optimal model minimises the total description length, which is equivalent to minimising the sum of bias (L_2) and variance (L_1).

3.9 Chapter Summary

MDL establishes compression as the unifying foundation of statistical learning:

- Two-part MDL subsumes MAP estimation and weight regularisation as special cases of minimising total description length (Proposition 3.2).
- The NML/stochastic complexity achieves minimax regret (Theorem 3.1) with complexity governed by the Fisher information volume (Theorem 3.2).
- BIC is the first-order MDL approximation (Corollary 3.1); full stochastic complexity is the ideal target.
- The ELBO of VAEs is the MDL objective for latent variable models (Theorem 3.4), establishing that VAE training is information-theoretically principled.
- MDL model selection is consistent (Theorem 3.5) and provides a rigorous information-theoretic formalisation of Occam's Razor.

Chapter 4

Mutual Information as a Diagnostic Tool in Deep Networks

Mutual information (MI) is the natural measure of statistical dependence between two random variables. For deep networks, it provides a coordinate-free diagnostic of what each layer has learned: I(T^ℓ; Y) measures how much the representation predicts the target, while I(X; T^ℓ) measures how much of the raw input is preserved. These quantities are invariant under bijective reparametrisations of the representation space, making them more principled than weight-norm or activation-norm statistics.

The central challenge is that MI is difficult to compute for high-dimensional continuous distributions. This chapter develops the main estimation strategies — neural estimators (MINE), k-NN estimators (KSG), and noise-injection approaches — and applies them as diagnostic tools for understanding saturation, dead neurons, neural collapse, and attention structure in transformers.

4.1 Why Mutual Information?

Consider two common alternatives to MI for measuring what a layer has learned:

- Weight norms $\|W_\ell\|$: coordinate-dependent (changes under reparametrisation of the weight space), does not directly measure information content.
- Activation variance $\mathrm{Var}(T^\ell)$: coordinate-dependent, confounds information content with the scale of activations.

MI avoids both problems: it is invariant under bijective transformations of T^ℓ and directly measures statistical dependence. However, it requires estimating joint and marginal distributions in high-dimensional spaces — a notoriously difficult problem.

The data processing inequality provides a fundamental constraint on MI across layers: for the Markov chain $Y \leftrightarrow X \to T^1 \to \dots \to T^L \to \hat{Y}$:

$$I(X;Y) \geq I(T^1;Y) \geq \dots \geq I(T^L;Y) \geq I(\hat{Y};Y),$$

$$I(X;T^1) \geq I(X;T^2) \geq \dots \geq I(X;T^L).$$

These monotonicity constraints mean that deeper layers cannot contain more information about Y than shallower layers, and cannot contain more information about X than shallower layers. The diagnostic question is: where in the network does information about Y become accessible (useful for prediction), and where does information about X become inaccessible (compressed away)?

4.2 The MINE Estimator

Theorem 4.1 (Donsker-Varadhan Representation).
For any two distributions P, Q on Z with P absolutely continuous with respect to Q:
$D_{KL}(P \| Q) = \sup_{f: Z \to \mathbb{R}} \{E_P[f] - \log E_Q[e^f]\},$

where the supremum is over all measurable functions $f : Z \to \mathbb{R}$. The supremum is achieved by $f^*(z) = \log(dP/dQ)(z) + \text{const}$.

Applying the DV representation to the joint-marginal decomposition $I(X;T) = D_{KL}(p(x,t) \,/\!/\, p(x)p(t))$ gives the MINE estimator.

Definition 4.1 (MINE Estimator, Belghazi et al. 2018).
Parametrise f by a neural network $f_\phi : X \times T \to \mathbb{R}$ (the critic network). For a batch of n paired samples $\{(x_i,t_i)\}$ and n independently drawn marginal samples $\{\tilde{t}_j\}$:
$\hat{I}_{MINE}(X;T;\phi) := (1/n) \sum_i f_\phi(x_i,t_i) - \log[(1/n) \sum_j e^{f_\phi(x_i,\tilde{t}_j)}]$.
The MINE estimate is $\hat{I}_{MINE} = \max_\phi \hat{I}_{MINE}(X;T;\phi)$, maximised via gradient ascent on ϕ.

MINE is a lower bound on $I(X;T)$ for any ϕ (Proposition 4.1), and it is tight in the limit of large n and infinite capacity of f_ϕ. The main practical limitation is bias: Jensen's inequality gives $\mathbb{E}[\log(1/n \sum e^f)] \geq \log \mathbb{E}[e^f]$, so the log-mean in MINE is a biased estimate of $\log \mathbb{E}[e^f]$. The bias is $O(1/n)$ but can be significant for small batch sizes.

Several corrected variants have been proposed:

- MINE-f: replaces the exponential with an f-divergence representation, reducing variance at the cost of increased bias.
- JS-MI (Nowozin et al. 2016): uses the Jensen-Shannon divergence representation, giving a bounded estimator with better variance properties.
- InfoNCE (van den Oord et al. 2018): uses a contrastive loss that is equivalent to a lower bound on $I(X;T)$ with contrastive samples.

4.3 The KSG Estimator for Moderate Dimensions

Theorem 4.2 (KSG Estimator, Kraskov et al. 2004).
For samples $\{(x_i,t_i)\}_i$ from $p(x,t)$, the KSG k-nearest-neighbour estimator is:
$\hat{I}_{KSG}(X;T) := \psi(k) + \psi(n) - \mathbb{E}_{x,t}[\psi(n_x) + \psi(n_t)]$,
where ψ is the digamma function, and n_x (resp. n_t) is the number of marginal x-neighbours (resp. t-neighbours) within the radius equal to the distance to the k-th nearest joint neighbour.
$\hat{I}_{KSG} \to I(X;T)$ in probability as $n \to \infty$ with $k = k(n) \to \infty$ and $k/n \to 0$. The bias is $O(k/n)$ and the variance is $O(1/k)$.

The KSG estimator is consistent and has no hyperparameters beyond k (the number of neighbours), making it simple to use in practice. Its main limitation is the curse of dimensionality: k-NN distances become uninformative in high-dimensional spaces. For activations with dimension $d > 20$, KSG estimates become unreliable, and MINE or PCA-based dimensionality reduction is recommended.

A practical guideline: use KSG when the activation dimension $d \leq 20$ and $n \geq 1000$; use MINE when $d > 20$ or when computational resources allow a neural critic. Both

estimators should be cross-validated: estimates should be stable across different choices of k (for KSG) or architecture (for MINE).

4.4 Noise Injection and the Stochastic Representation Framework

The fundamental obstacle to MI computation for deterministic networks is Proposition 1.1: I(X; f(X)) = +∞ for continuous X and injective f. The standard resolution is to add noise to the activations, making the map stochastic.

Definition 4.2 (Gaussian Noise Injection).
For a deterministic layer T^ℓ = f_ℓ(T^{ℓ-1}), the noise-injected representation is:
T^ℓ_σ = f_ℓ(T^{ℓ-1}) + ε, ε ~ N(0, σ²I_{d_ℓ}),
with stochastic encoder q_σ(t^ℓ | t^{ℓ-1}) = N(t^ℓ; f_ℓ(t^{ℓ-1}), σ²I). The noise-regularised MI is:
I_σ(X; T^ℓ) := I(X; T^ℓ_σ) = H_σ(T^ℓ) − H_σ(T^ℓ | X),
which is finite for any σ > 0 and equals (d_ℓ/2) log(1/σ²) + O(1) as σ → 0 (see Proposition 5.1).

Dropout (Srivastava et al. 2014) is a structured noise injection with Bernoulli-distributed zeroing: T^ℓ_j → T^ℓ_j · Bernoulli(1−p). Its effect on I(X; T^ℓ) is to reduce it by approximately p · H(T^ℓ_j | X) for each dropped neuron j — the information that the dropped neuron carried about X that is not redundantly encoded by other neurons.

4.5 MI as a Diagnostic: Saturation, Dead Neurons, and Collapse

4.5.1 Saturation and Information Loss

Proposition 4.1 (Saturation Reduces MI).
Let T^ℓ_j = σ(w^T_j T^{ℓ-1}) be a neuron with sigmoid activation σ. If |w^T_j T^{ℓ-1}| > C with probability ≥ 1 − ε (the neuron is saturated with probability 1−ε), then:
I(X; T^ℓ_j) ≤ H(ε) + ε log |T^ℓ_j|,
where H(ε) = −ε log ε − (1−ε) log(1−ε). For ε → 0, I(X; T^ℓ_j) → 0.
Saturation causes information loss: a saturated neuron outputs ≈ 0 or ≈ 1 regardless of the input, carrying no information about X.

The information-theoretic rationale for batch normalisation and layer normalisation follows directly: by controlling the scale of pre-activations, normalisation prevents saturation and thereby maintains I(X; T^ℓ) at a healthy level throughout training. Empirically, layers without normalisation in deep networks often exhibit saturation-induced MI collapse after many training steps.

4.5.2 Dead ReLU Units

A ReLU neuron j is dead if w^T_j T^{ℓ-1} ≤ 0 for all x in the training distribution. Dead neurons contribute T^ℓ_j = 0 almost surely, so I(X; T^ℓ_j) = 0. If a fraction ϱ of

neurons in layer ℓ are dead, the effective information capacity of the layer is reduced by the fraction of dead neurons:

$$I(X;\, T^\ell) \leq I(X;\, T^\ell_{alive}),$$

where T^ℓ_{alive} is the sub-vector of non-dead neurons. This provides an information-theoretic upper bound on the utility of a layer with many dead neurons.

The practical diagnostic protocol: monitor both (i) the dead neuron fraction ρ^ℓ (the fraction of neurons that never activate on any training example) and (ii) the estimated MI $I_\sigma(T^\ell; Y)$. A layer with high ρ^ℓ and low $I_\sigma(T^\ell; Y)$ is both structurally and informationally impoverished. The solution is to use initialisation strategies (He initialisation, Xavier initialisation) and optimisers (Adam, RMSProp) that prevent dead neurons during the early phases of training.

4.5.3 Neural Collapse and the Terminal Phase of Training

Neural Collapse (Papyan et al. 2020) describes the phenomenon observed during the terminal phase of training on balanced classification tasks: the within-class variance of last-layer features collapses to zero, the class mean vectors $\mu_1, \ldots, \mu_C$ form a simplex equiangular tight frame (ETF), and the linear classifier aligns with the class means.

Proposition 4.2 (Neural Collapse and MI).
In the neural collapse regime, the last-layer representation T^L satisfies:
$I(T^L; X) \approx \log C$ *(C-point representation, one per class),*
$I(T^L; Y) \approx H(Y)$ *(perfect class prediction under balanced classes).*
The representation has discarded all within-class variation in X while perfectly retaining class label information — the IB fixed point at $\beta \to \infty$.

Neural collapse is therefore the information-theoretically optimal terminal state of classifier training: maximum compression ($I(T^L;X) = \log C$ is the minimum possible) consistent with perfect prediction ($I(T^L;Y) = H(Y)$). Its onset can be detected by monitoring $I_\sigma(T^L; X)$: a sharp decrease toward $\log C$ during terminal training is the MI signature of collapse.

4.6 The Information Plane: A Corrected Interpretation

Definition 4.3 (Noise-Regularised Information Plane).
For fixed noise level $\sigma > 0$, the noise-regularised information plane is the set of layer coordinates:
$P_\sigma := \{(I_\sigma(X; T^\ell), I_\sigma(T^\ell; Y)) : \ell = 1, \ldots, L\},$
where I_σ denotes MI computed under Gaussian noise injection at level σ.
The data processing inequality gives the monotonicity constraints:
$I_\sigma(X; T^\ell) \geq I_\sigma(X; T^{\ell+1})$ *and* $I_\sigma(T^\ell; Y) \geq I_\sigma(T^{\ell+1}; Y)$.

Under this definition, the quantities are finite and well-behaved, and their dynamics during training can be meaningfully interpreted. The key insight is that the noise level σ

must be fixed throughout training: comparing I_σ at different time steps with different σ values gives spurious dynamics.

Current empirical evidence suggests that the dynamics in P_σ are highly architecture- and noise-level-dependent:

- For tanh networks with small σ: approximate two-phase dynamics can appear, but they are artefacts of the near-saturation regime and the specific noise level.
- For ReLU networks with small σ: both I_σ(X;T^ℓ) and I_σ(T^ℓ;Y) increase monotonically during training, with no compression phase.
- For ResNet-class architectures: the information plane dynamics are complex, with different layers exhibiting different behaviours depending on the skip connections.

4.7 Practical MI Diagnostics: A Toolkit

4.7.1 Centred Kernel Alignment (CKA)

Definition 4.4 (Linear CKA, Kornblith et al. 2019).
Let $T^\ell \in R^{n\times d_\ell}$ be the matrix of layer-ℓ activations on a batch of n examples, and let $K = T(T^\ell)^T$, $L = (T^{\ell'})^T T^{\ell'}$ be the Gram matrices. Linear CKA is:
$CKA(K,L) := \| L^T K \|^2_F / (\| K \|_F \| L \|_F)$.
CKA = 0 for independent representations, CKA = 1 for representations related by an orthogonal transformation. CKA is invariant under isotropic scaling and orthogonal transformations of each representation independently.

CKA provides a practical alternative to MI estimation when the goal is to compare representations across layers or models. Its computational cost is $O(n^2 d)$ per layer pair, which is feasible for batch sizes $n \le 10^4$. CKA has been used to:

- Compare representations across different architectures (ResNet vs. VGG vs. ViT).
- Identify which layers are most similar between two trained networks with the same architecture.
- Detect representational collapse (a layer converging to a low-rank representation that is nearly identical for all inputs).

4.7.2 V-Usable Information

V-usable information (Xu et al. 2020) provides a computable, model-class-constrained analogue of MI:

$$I_V(X \to Y) := H(Y) - \inf_{v \in V} E_{p(x,y)}[-\log v(y|x)].$$

$I_V(T^\ell \to Y)$ with V = linear classifiers is the amount of class-predictive information that is linearly separable in layer ℓ. Comparing $I_V(T^\ell \to Y)$ across layers identifies the layer where the non-linear transformation into linear separability occurs — typically the penultimate layer for well-trained classifiers.

4.8 Information-Theoretic Diagnostics for Transformers

4.8.1 Attention Entropy

For a transformer with attention matrices A^h ∈ R^{L×L} (one per head h, one row per query position), the attention entropy of head h is:

$$H_h := -(1/L) \Sigma_i \Sigma_j A^h[i,j] \log A^h[i,j].$$

High H_h (diffuse attention, all attention weights approximately uniform) corresponds to a layer that aggregates information broadly from all input positions. Low H_h (sharp attention, most weight on a few positions) corresponds to localised information retrieval.

Empirically, attention heads in pre-trained language models exhibit a variety of specialised behaviours identifiable by their attention entropy:

- Syntactic heads (H_h ≈ 0.5–1.5 bits): sharp attention to syntactically related tokens (subject-verb pairs, determiners and nouns).
- Positional heads (H_h ≈ 0): always attend to a fixed relative position (previous token, next token).
- Global heads (H_h ≈ log L): diffuse attention to all tokens, producing a global summary of the sequence.
- [CLS] heads: attention concentrated on the special [CLS] token, aggregating global context.

4.8.2 Head Redundancy via CKA

Applying CKA between the output representations of attention heads h and h' within the same layer:

$$R(h, h') := CKA(T^{\ell,h}, T^{\ell,h'}),$$

quantifies redundancy: R(h,h') ≈ 1 means the two heads carry nearly identical information. Michel et al. (2019) showed that for BERT-base, 70% of heads can be pruned with < 1% degradation in downstream accuracy, suggesting high CKA-redundancy among heads. The information-theoretically principled pruning criterion is: remove head h if max_{h'≠h} R(h,h') > threshold.

4.9 Chapter Summary

MI is the ideal diagnostic for deep network representations, but its computation requires careful treatment:

- MINE (Theorem 4.1): neural lower bound via Donsker-Varadhan, suitable for high-dimensional activations.
- KSG (Theorem 4.2): consistent k-NN estimator for moderate-dimensional activations (d ≤ 20).
- Noise injection (Definition 4.2): makes MI well-defined for deterministic layers.

- Saturation, dead neurons, and neural collapse all produce characteristic MI signatures (Propositions 4.1, 4.2).
- CKA and V-usable information provide practical alternatives when full MI estimation is intractable.
- Attention entropy and CKA-redundancy characterise information structure in transformer heads.

Chapter 5

Geometric Measure Theory and Representation Learning

The geometric perspective on representation learning asks: what is the shape of the data distribution, and how does this shape constrain what networks can learn? High-dimensional data — images, audio, text — does not fill R^D uniformly. It concentrates near low-dimensional manifolds embedded in the ambient space. This manifold structure is the key to understanding why networks trained on D-dimensional inputs can generalise from $n \ll D^D$ samples.

Geometric measure theory (GMT) provides the mathematical tools to formalise this intuition: Hausdorff dimension, rectifiable sets, the coarea formula, and optimal transport. These tools are not merely ornamental — they are needed to make precise statements about the information content of representations, the geometry of compression, and the relationship between intrinsic dimension and sample complexity.

5.1 Hausdorff Dimension and the Effective Dimension of Data

Definition 5.1 (Hausdorff Dimension).
For a set $E \subseteq R^D$, the s-dimensional Hausdorff measure is $H^s(E) = \lim_{\delta \to 0} \inf\{\sum_i (\mathrm{diam}\, U_i)^s : E \subseteq \bigcup_i U_i, \mathrm{diam}\, U_i \le \delta\}$.
The Hausdorff dimension is $\dim_H(E) = \inf\{s : H^s(E) = 0\} = \sup\{s : H^s(E) = +\infty\}$.
For smooth d-dimensional submanifolds of R^D: $\dim_H = d$ and H^d coincides with the Riemannian volume measure.

Proposition 5.1 (MI and Hausdorff Dimension).
Let X be supported on $E \subseteq R^D$ with $\dim_H(E) = d$. Let $T = f(X) + \varepsilon$ with $\varepsilon \sim N(0, \sigma^2 I_D)$. Then:
$I_\sigma(X;T) = (d/2)\log(1/\sigma^2) + O(1)$ as $\sigma \to 0$.
The slope of $I_\sigma(X;T)$ vs. $\log(1/\sigma)$ estimates the intrinsic dimension d, independent of ambient dimension D.

This result is fundamental: it connects MI estimation to intrinsic dimensionality estimation. The slope of $\log I_\sigma$ vs. $\log(1/\sigma)$ can be estimated from data, providing a non-parametric estimator of the data manifold dimension. Networks that compress X to a representation T with $\dim_H(\mathrm{supp}\ T) < \dim_H(\mathrm{supp}\ X)$ are performing genuine dimensional reduction, not just coordinate transformation.

Corollary 5.1 (Intrinsic Dimension Profile of a Network).
The sequence $d^{(\ell)}_H = \dim_H(\mathrm{supp}\ p(t^\ell))$ across layers $\ell = 0, \ldots, L$ forms the network's intrinsic dimension profile. For well-trained classifiers:
$d^{(0)}_H = \dim_H(\text{data manifold})$ (typically 20–100 for natural images),
$d^{(L)}_H \le \log_2 C$ (at most $\log C$ for C-class classification).

Intermediate layers may exhibit a transient expansion ($d^{(\ell)}_H > d^{(0)}_H$) in the 'unfolding' phase before collapsing.

5.2 Rectifiable Sets and Network Layer Maps

Theorem 5.1 (Lipschitz Maps Preserve Rectifiability).
If $E \subseteq R^{d_{\ell-1}}$ is countably d-rectifiable and $f^\ell : R^{d_{\ell-1}} \to R^{d_\ell}$ is Lipschitz with constant L_f, then $f^\ell(E)$ is countably d-rectifiable and $H^d(f^\ell(E)) \leq L_f^d \cdot H^d(E)$.
Consequence: ReLU networks (piecewise linear, hence Lipschitz on compact domains) map rectifiable data distributions to rectifiable representations. The Hausdorff measure can only decrease (or stay constant for bijective maps). The sequence $d^{(\ell)}_H$ is non-increasing.

The non-increasing property of intrinsic dimension across layers is a key geometric consequence of the DPI: deeper layers cannot have higher intrinsic dimension than shallower layers (because intrinsic dimension is an information quantity, and the DPI says information can only decrease along a Markov chain). However, the local intrinsic dimension can transiently increase if the network unfolds the data manifold — stretching it in some directions while contracting others.

5.3 The Coarea Formula and Information Decomposition

Theorem 5.2 (Coarea Formula, Federer 1959).
Let $f : R^D \to R^d$ ($D \geq d$) be Lipschitz and $g : R^D \to [0,\infty)$ measurable. Then:
$$\int_{R^D} g(x)\, J_d f(x)\, dH^D(x) = \int_{R^d} \left[\int_{f^{-1}(t)} g(x)\, dH^{D-d}(x)\right] dH^d(t),$$
where $J_d f(x) = \sqrt{\det(\nabla f(x) \nabla f(x)^T)}$ is the d-Jacobian of f.

The coarea formula decomposes integration over the domain into integration over the codomain weighted by the fibre measure. In the information-theoretic context, the fibre $f^{-1}(t)$ is the set of all inputs that map to the same representation t. Large fibres mean many inputs map to the same representation — this is compression. The fibre volume $H^{D-d}(f^{-1}(t))$ measures the amount of information about X that is lost when X is mapped to $T = f(X)$.

Proposition 5.2 (Compression Decomposition via Coarea).
The compression $H(X) - H(T^\ell)$ decomposes into two parts:
$H(X) - H(T^\ell) =$ *[isometric compression] + [destructive compression],*
where isometric compression $= E_{p(x)}[\log J_d f^\ell(x)]$ (invertible stretching/squeezing that changes coordinates but preserves information) and destructive compression $= E_{p(t^\ell)}[\log H^{D-d}((f^\ell)^{-1}(t^\ell))]$ (true information erasure by collapsing fibres).
An ideal IB-optimal layer has zero destructive compression in the relevant directions and controlled isometric compression that maps the data manifold to a simpler coordinate system.

5.4 Optimal Transport and Representation Comparison

Definition 5.2 (Wasserstein-2 Distance).
For probability measures μ, ν on $\mathbb{R}^d$ with finite second moments, the Wasserstein-2 distance is:
$W_2(\mu,\nu) := \left(\inf_{\gamma\in\Pi(\mu,\nu)} \int \|x-y\|^2 \, d\gamma(x,y)\right)^{1/2}$,
where $\Pi(\mu,\nu)$ is the set of couplings with marginals μ and ν.
For Gaussians: $W_2^2(N(m_1,\Sigma_1), N(m_2,\Sigma_2)) = \|m_1-m_2\|^2 + \mathrm{tr}(\Sigma_1+\Sigma_2-2(\Sigma_1^{1/2}\Sigma_2\Sigma_1^{1/2})^{1/2})$.

The Wasserstein-2 distance has key advantages over KL-based divergences for comparing representations:

- W_2 is defined even when μ and ν have disjoint supports (as happens for representations from different training checkpoints or architectures), while $D_{KL}(\mu \| \nu) = +\infty$ when $\mu \not\ll \nu$.
- W_2 metrises the space of probability distributions with the geometry of optimal transport: the geodesic between μ and ν is a straight-line interpolation in the space of distributions, corresponding to linear interpolation of the optimal transport map.
- W_2 is continuous with respect to weak convergence of distributions, making it robust to approximation errors.

Theorem 5.3 (Stability of Representations under Weight Perturbation).
Let f_w and $f_{w'}$ be two networks with $\|w - w'\| \le \varepsilon$ (small weight perturbation). Then:
$W_1(f_w\#p(x), f_{w'}\#p(x)) \le Lip(f_w - f_{w'}) \cdot W_1(p(x), p(x)) = 0$? *No:*
$W_1(f_w\#p(x), f_{w'}\#p(x)) \le \sup_x \|f_w(x) - f_{w'}(x)\| \le L_f \cdot \varepsilon$,
where L_f is the Lipschitz constant of the network. Small weight perturbations produce small (W_1-bounded) changes in the representation distribution.

5.5 Intrinsic Dimension Estimation and the Correlation Dimension

Definition 5.3 (Correlation Dimension, Grassberger-Procaccia 1983).
For a sample $\{t_1, \ldots, t_n\} \subseteq \mathbb{R}^d$, the correlation integral is $C(r) = (2/n(n-1)) \sum_{i<j} 1[\|t_i - t\| \le r]$.
The correlation dimension is $d_{corr} := \lim_{r\to 0} \log C(r) / \log r$.
For a d-dimensional smooth manifold: $d_{corr} = d$ exactly. For fractal sets: $d_{corr} = \dim_H$.

Empirical studies (Ansuini et al. 2019; Gong et al. 2019) of intrinsic dimension profiles for ResNet, VGG, and BERT:

- ResNet-50: intrinsic dimension peaks at ~100 in intermediate layers, drops to ~10 in the penultimate layer, and reaches ~7 ($\log_2 1000 \approx 10$ for ImageNet with 1000 classes) at the final layer.

- BERT-large: intrinsic dimension peaks at ~30–40 in middle layers (around layers 9–12) and drops to ~10–15 at the output. The peak corresponds to the 'contextual integration' layers where long-range dependencies are captured.
- VGG-16: monotone decrease in intrinsic dimension throughout the network, from ~100 at the first convolutional layer to ~10 at the penultimate fully-connected layer.

5.6 Covering Numbers and Dimension-Dependent Generalisation

Theorem 5.4 (Dudley's Entropy Integral Bound).
The Rademacher complexity of F satisfies:
$R_n(F) \leq \inf_{\varepsilon>0} [4\varepsilon + (12/\sqrt{n}) \int_{\varepsilon}^{diam(F)} \sqrt{(\log N(F, \|\cdot\|_{L^2}, u))}\, du]$.
For Lipschitz networks on a d-dimensional data manifold with ambient dimension D:
$R_n(F) = O(L_f \cdot \sqrt{(d \log(L_f\, diam(data)/(\varepsilon^2)) / n))}$,
independent of D. The generalisation bound depends on d, not D.

Corollary 5.2 (Dimension-Dependent Generalisation Bound).
Let X be supported on a d-dimensional submanifold of R^D, and F a Lipschitz network class with constant L_f. With probability $\geq 1 - \delta$:
$L(f) \leq \hat{L}(f) + O(L_f \sqrt{(d \log(n\, L_f^2/\delta) / n))}$.
This bound is independent of the ambient dimension D — it depends only on the intrinsic dimension d of the data support.

5.7 Chapter Summary

The GMT framework reveals the geometric substrate of representation learning:

- Hausdorff dimension controls noise-regularised MI (Proposition 5.1), making MI a consistent intrinsic dimension estimator.
- Rectifiable network maps preserve rectifiability with non-increasing intrinsic dimension (Theorem 5.1).
- The coarea formula decomposes compression into isometric and destructive parts (Proposition 5.2).
- Wasserstein distances provide stable, well-behaved metrics for representation comparison (Theorem 5.3).
- Empirical intrinsic dimension profiles show transient expansion then collapse to log C (Corollary 5.1).
- Dimension-dependent generalisation bounds confirm that the effective sample complexity is determined by intrinsic dimension d, not ambient dimension D (Corollary 5.2).

Chapter 6

Rate-Distortion Theory and Generative Models

Shannon's rate-distortion theory provides the fundamental limits of lossy compression: given a source distribution and a distortion measure, what is the minimum description rate needed to achieve a given average distortion? This chapter develops rate-distortion theory and applies it to generative modelling, establishing that VAEs, GANs, diffusion models, and normalizing flows can all be understood as parametric approximations to points on the rate-distortion curve of the data distribution.

6.1 The Rate-Distortion Problem

Definition 6.1 (Rate-Distortion Function).
For source $X \sim p(x)$ and distortion measure $d : X \times \hat{X} \to [0,\infty)$, the rate-distortion function is:
$R(D) := \min_{p(\hat{x}|x):\, \mathbb{E}[d(X,\hat{X})] \le D} I(X;\hat{X})$.
$R(D)$ is the minimum MI between source and reconstruction subject to average distortion $\le D$.
$R(D)$ is non-increasing, convex, and equals $H(X)$ at $D=0$ (lossless) and 0 at $D=D_{max}$.

Shannon's achievability theorem states that for any $\varepsilon > 0$, there exist block codes of length n that achieve distortion $D + \varepsilon$ at rate $R(D) + \varepsilon$ for n large enough. The converse states that no code can achieve distortion D at rate $< R(D)$. Together, these theorems establish $R(D)$ as the fundamental limit of lossy compression.

The connection to IB: the IB of Chapter 1 is the rate-distortion problem with $\hat{X} = T$ and distortion measure $d(x,t) = D_{KL}(p(y|x) \,/\!/\, p(y|t))$. The IB curve C_{IB} is the rate-distortion curve of $p(x)$ under this specific distortion measure.

6.2 The Blahut-Arimoto Algorithm

Theorem 6.1 (Blahut-Arimoto).
For discrete sources, the rate-distortion function $R(D)$ is computed by alternating:
$q(\hat{x}) \leftarrow \sum_x p(x)\, p(\hat{x}|x)$,
$p(\hat{x}|x) \leftarrow q(\hat{x}) \exp(s \cdot d(x,\hat{x})) \,/\, Z(x,s)$, $\quad Z(x,s) = \sum_{\hat{x}'} q(\hat{x}') \exp(s \cdot d(x,\hat{x}'))$,
where $s \le 0$ is the Lagrange multiplier ($s = -\beta$ in IB notation). The algorithm decreases the Lagrangian $I(X;\hat{X}) + s \cdot \mathbb{E}[d(X,\hat{X})]$ at each step and converges to the global optimum at linear rate.

The neural analogue: the VIB training loop (Chapter 1, Section 1.7.2) is precisely the Blahut-Arimoto algorithm applied to a continuous, high-dimensional source, with:

- The encoder $q_\phi(t|x)$ playing the role of the conditional $p(\hat{x}|x)$.
- The aggregate posterior $q_\phi(t) = \mathbb{E}_{p(x)}[q_\phi(t|x)]$ playing the role of $q(\hat{x})$.

- The ELBO gradient alternating between encoder and decoder updates — a stochastic, continuous-variable generalisation of the BA alternating updates.

6.3 VAEs and the Rate-Distortion Curve

The β-VAE (Higgins et al. 2017) parametrises the rate-distortion trade-off explicitly:

$$L_{\beta\text{-}VAE}(\phi,\theta) = E_{p(x)}\, E_{q_\phi(z|x)}[-\log p_\theta(x|z)] + \beta \cdot E_{p(x)}[D_{KL}(q_\phi(z|x) \,\|\, p(z))].$$

Theorem 6.2 (β-VAE Traces the Rate-Distortion Curve).
As β varies from 0 to ∞, the optimal $(R^*(\beta), D^*(\beta))$ pair:
$R^*(\beta) := E_{p(x)}[D_{KL}(q^*_\phi(z|x) \,\|\, p(z))]$, $D^*(\beta) := E_{p(x)}[-E_{q^*_\phi(z|x)}[\log p^*_\theta(x|z)]]$,
traces a path along the rate-distortion curve $C_{RD}(p(x))$. In the limit of infinite model capacity: $\{(R^*(\beta), D^*(\beta)) : \beta \geq 0\} = C_{RD}(p(x))$.
High $\beta \to$ low rate (compressed latent codes) and high distortion (blurry reconstructions).
Low $\beta \approx 0 \to$ high rate (large latent codes) and low distortion (sharp reconstructions).

6.4 The Gaussian Rate-Distortion Function and Water-Filling

Theorem 6.3 (Gaussian Water-Filling).
For $X \sim N(0,\Sigma)$ with eigenvalues $\sigma^2_1 \geq \ldots \geq \sigma^2_D$ and squared-error distortion, the rate-distortion function is:
$R(D) = \sum_i (1/2) \max\{0, \log(\sigma^2_i/\lambda)\}$,
where $\lambda > 0$ is the water-filling level satisfying $\sum_i \min(\sigma^2_i,\lambda) = D$.
The optimal encoder discards the directions with $\sigma^2_i \leq \lambda$ (not enough 'signal' to justify the communication cost) and encodes the remaining directions at rates $(1/2)\log(\sigma^2_i/\lambda) \geq 0$.

The water-filling interpretation: think of the eigenvalues σ^2_i as 'water levels' and λ as the 'water surface'. Directions where $\sigma^2_i > \lambda$ are 'above water' and receive positive rate. Directions where $\sigma^2_i \leq \lambda$ are 'underwater' and receive zero rate (discarded). As D increases (more distortion allowed), λ rises and fewer directions are above water.

Deep learning implications:

- PCA-based preprocessing: selecting the top-d principal components is approximately optimal for Gaussian data (the optimal encoder spans the top-d eigenvectors of Σ).
- Diffusion models: the noise schedule $\beta(t)$ determines which frequency bands of the data receive positive 'rate' (are learned by the denoiser). Optimal schedules allocate bits to the most informative frequency bands.
- Representation learning: the directions that VAEs learn to encode (high-variance directions in latent space) correspond to the high-eigenvalue directions of the data covariance.

6.5 GANs and the Primal-Dual Rate-Distortion Problem

Theorem 6.4 (WGAN and W_1 Minimisation).
The Wasserstein GAN objective with 1-Lipschitz discriminator:
$\min_G \max_{\|D\|_{Lip} \le 1} \mathbb{E}_{p(x)}[D(x)] - \mathbb{E}_{p(z)}[D(G(z))]$,
equals $W_1(p(x), G\#p(z))$ by the Kantorovich-Rubinstein duality theorem.
At optimum, the generator G^* satisfies $G^*\#p(z) = p(x)$ — it produces samples from the true data distribution.

The rate-distortion perspective on GANs: GANs minimise distortion (in the W_1 metric) without a rate constraint. The latent code $z \in \mathbb{R}^k$ carries $H(p(z)) = \infty$ bits of information (for continuous z), and there is no encoder that maps x to z — the GAN objective does not define an inference mechanism. This explains the fundamental difference between VAEs (which have an explicit encoder and rate constraint) and GANs (which do not).

6.6 Diffusion Models as Multi-Scale Rate-Distortion Coders

Theorem 6.5 (Diffusion ELBO as Rate-Distortion).
The negative ELBO of a diffusion model with T noise scales:
$-L_{ELBO} = \mathbb{E}[-\log p_\theta(x_0|x_1)] + \sum_{t=2}^{T} \mathbb{E}[D_{KL}(q(x_{t-1}|x_t,x_0) \| p_\theta(x_{t-1}|x_t))] + D_{KL}(q(x_T|x_0) \| p(x_T))$.
Each KL term $\mathbb{E}[D_{KL}(\dots)]$ at scale t is the rate-distortion cost of encoding x_0 at noise scale t: the amount of information about x_0 that the denoiser at scale t fails to capture.

The diffusion model is a hierarchical rate-distortion coder: it allocates information across T noise scales, with fine-grained details (high-frequency structure) encoded at small t (low noise) and coarse structure encoded at large t (high noise). The optimal noise schedule $\beta^*(t)$ corresponds to the water-filling solution applied to the Fourier decomposition of the data covariance.

6.7 The Perception-Distortion Trade-off

Theorem 6.6 (Perception-Distortion Impossibility, Blau-Michaeli 2018).
For any estimator $\hat{X} = g(Y)$ of X from degraded Y, there exists a constant $c > 0$ such that:
$\mathcal{L}(p(\hat{x}), p(x)) + c \cdot \mathbb{E}[d(X, \hat{X})] \geq \text{const}$,
where $\mathcal{L}$ is any perceptual quality metric (Fréchet Inception Distance, LPIPS). Perfect perceptual quality ($\mathcal{L} = 0$) and minimum MSE distortion are mutually exclusive for most ill-posed inverse problems.

This theorem explains a ubiquitous empirical phenomenon: MSE-trained image restoration networks produce blurry outputs (low MSE but poor perceptual quality), while GAN-trained networks produce sharp outputs (good perceptual quality but higher MSE). The trade-off is not a failure of optimisation but a fundamental

information-theoretic impossibility. Any estimator that achieves $\Delta = 0$ must introduce stochasticity (to sample from $p(x)$ rather than collapse to $E[X|Y]$), which necessarily increases MSE.

6.8 Chapter Summary

- $R(D)$ is the minimum rate to achieve distortion D; it equals the IB curve with the KL-divergence distortion measure (Definition 6.1).
- Blahut-Arimoto computes $R(D)$ iteratively; VIB training is its neural analogue (Theorem 6.1).
- β-VAE traces the rate-distortion curve as β varies (Theorem 6.2).
- Water-filling allocates bits to high-variance directions and discards low-variance ones (Theorem 6.3).
- GANs minimise W_1-distortion without a rate constraint, explaining their absent inference mechanism (Theorem 6.4).
- Diffusion models are multi-scale rate-distortion coders (Theorem 6.5).
- Perception-distortion trade-off is a fundamental impossibility (Theorem 6.6).

Chapter 7

Information-Theoretic Lower Bounds on Sample Complexity

Upper bounds tell us what is achievable. Lower bounds tell us what is impossible. This chapter develops the three principal information-theoretic lower bound techniques — Fano's inequality, Le Cam's method, and Assouad's lemma — and applies them to obtain tight lower bounds on the sample complexity of deep learning, transfer learning, and federated learning.

7.1 Fano's Inequality and Hypothesis Testing Lower Bounds

Theorem 7.1 (Fano's Inequality).
Let M be uniform over $M = \{1,\dots,m\}$, and $\hat{M}(Y)$ any estimator with $M \to Y \to \hat{M}$ a Markov chain. Then:
$P(\hat{M} \neq M) \geq 1 - (I(M;Y) + \log 2) / \log m.$
Equivalently, $I(M;Y) \geq (1 - P_e) \log m - \log 2$, where $P_e = P(\hat{M} \neq M)$.

Fano's inequality is the cornerstone of minimax lower bounds in statistics. The standard workflow:

7. Construct m hypotheses $\theta_1, \dots, \theta_m$ that are (i) well-separated in the estimation metric $d(\theta_i, \theta) \geq \alpha$ and (ii) hard to distinguish from n i.i.d. observations (small $I(M; X_1^n)$).
8. Bound $I(M; X_1^n) \leq n \cdot I(M; X_1)$ by the chain rule for i.i.d. data.
9. Apply Fano: $P_e \geq 1 - (n \cdot I(M;X_1) + \log 2) / \log m$.
10. Convert error probability to risk: $\mathbb{E}[d(\hat{\theta},\theta)] \geq (\alpha/2) \cdot P_e$.

Example application to mean estimation: $X_1, \dots, X_n \sim N(\mu, \sigma^2)$, $\mu \in [-B, B]$. Choose $m = e^{n\delta^2/(8\sigma^2)}$ hypotheses $\mu_1, \dots, \mu_m$ spaced 2δ apart. Then $I(M; X_1) \leq \delta^2/(2\sigma^2)$. Fano gives $P_e \geq 1/2$ if $\delta = \sigma/\sqrt{n}$. Converting: $\mathbb{E}[(\hat{\mu} - \mu)^2] \geq \sigma^2/(4n)$, matching the Cramér-Rao lower bound.

7.2 Le Cam's Two-Point Method

Theorem 7.2 (Le Cam's Method).
For two hypotheses θ_0, θ_1 with associated product distributions $P_0 = p_0^n$, $P_1 = p_1^n$:
$\max_{j \in \{0,1\}} \mathbb{E}_{P_j}[d(\hat{\theta}, \theta_j)] \geq (d(\theta_0,\theta_1)/2) \cdot (1 - \| P_0 - P_1 \|_{TV}).$
Using $TV \leq \sqrt{(D_{KL}(P_0 \| P_1)/2)} \leq \sqrt{(n\, D_{KL}(p_0 \| p_1)/2)}$, the minimax risk is bounded away from zero if $n \cdot D_{KL}(p_0 \| p_1) < 2$.

Le Cam's method is simpler than Fano but applies only to two hypotheses. It is most useful when the two hypotheses are naturally identified (e.g., null and alternative in hypothesis testing) and when bounding D_{KL} is easier than constructing a large packing.

7.3 Assouad's Lemma for Multivariate Parameters

Theorem 7.3 (Assouad's Lemma).
Let $\Theta = \{-1,+1\}^d$ and $\{P_\theta\}$ a family with (i) $d(\theta,\theta') \geq a$ whenever θ,θ' differ in coordinate j and (ii) $D_{KL}(P_\theta \| P_{\theta^{\oplus j}}) \leq \beta$ for all θ. Then:
$\max_\theta E[d(\hat{\theta},\theta)] \geq (ad/2)(1 - \sqrt{(\beta/2)})$.
For i.i.d. observations with n samples per coordinate: $\beta = n \cdot D_{KL}(P_0 \| P_1)$ for each coordinate, giving minimax risk $\geq (ad/2)(1 - \sqrt{(n \cdot D_{KL}/2)})$.

Corollary 7.1 (Minimax Rate for Nonparametric Regression).
For $f : [0,1]^d \to \mathbb{R}$ in the s-Hölder class H^s_d from n i.i.d. noisy observations:
$\inf_{\hat{f}} \sup_{f \in H^s_d} E[\| \hat{f} - f \|^2_{L^2}] = \Theta(n^{-2s/(2s+d)})$.
The rate degrades as d increases (curse of dimensionality) and improves as s increases (smoothness bonus).

7.4 Deep Networks Circumvent the Curse for Compositional Functions

Theorem 7.4 (Schmidt-Hieber 2020 — Deep Network Minimax Rate).
For $(d_1,\ldots,d_L,s)$-compositional functions $f = g_L \circ \cdots \circ g_1$ with effective dimension $\tilde{d} = \max_\ell d_\ell$:
Minimax rate $= \Theta(n^{-2s/(2s+\tilde{d})})$,
achieved by ReLU networks with depth $\Theta(L)$ and width $\Theta(n^{\tilde{d}/(2s+\tilde{d})})$.
For compositional functions with $\tilde{d} \ll d$, the rate $n^{-2s/(2s+\tilde{d})}$ is exponentially better than the non-compositional rate $n^{-2s/(2s+d)}$.

The deep network minimax rate is the information-theoretic justification for why deep networks are necessary for learning compositional functions. A shallow network of equivalent width would need exponentially more samples to achieve the same rate, because it cannot exploit the hierarchical structure of the target function.

Example: if $f = g_3(g_2(g_1(x)))$ with $d = 100$ (ambient dimension), $d_1 = d_2 = d_3 = 5$ (all intermediate dimensions ≤ 5), and $s = 2$ (twice-differentiable), then:

- Non-compositional rate: $n^{-2\cdot2/(2\cdot2+100)} = n^{-4/104} \approx n^{-0.038}$. Need $n \approx 10^{27}$ samples for 10% error.
- Compositional rate: $n^{-2\cdot2/(2\cdot2+5)} = n^{-4/9} \approx n^{-0.44}$. Need $n \approx 10^5$ samples for 10% error.

The 22-order-of-magnitude difference in sample complexity demonstrates why depth matters for high-dimensional compositional problems.

7.5 MI Generalisation Bounds and the IZS Method

Theorem 7.5 (Xu-Raginsky Bound).
For any learning algorithm $A : S \to H$ and σ-subgaussian loss ℓ:
$E[L(A(S)) - \hat{L}(A(S))] \leq \sqrt{(2\sigma^2 I(A(S);S) / n)}$,

where $I(A(S);S)$ is the MI between the learned hypothesis and the training set.

Corollary 7.2 (Compression Implies Generalisation).
If $I(A(S);S) \leq C$, the expected generalisation gap $\leq \sqrt{(2\sigma^2 C/n)}$. Algorithms that memorise training data ($I = \Omega(n)$) have generalisation gap $\Omega(\sigma\sqrt{2})$.

Theorem 7.6 (CMI Bound, Steinke-Zakynthinou 2020).
For the supersample $\tilde{U} = (Z_1,\ldots,Z_{2n})$ and training selection $U \in \{0,1\}^n$:
$CMI(A) := I(A(S,\tilde{U});U \mid \tilde{U}) \leq I(A(S);S)$.
$E[L(A) - \hat{L}(A)] \leq \sqrt{(2\sigma^2\, CMI(A)/n)}$.
The CMI bound is tighter: CMI captures whether the model overfits to the specific training partition, not just whether it memorises data values.

7.6 Lower Bounds for Transfer Learning

Theorem 7.7 (Transfer Learning Lower Bound).
For source D_S and target D_T with $D_{KL}(D_T \| D_S) = \varkappa$, any learner using m source and n target samples needs:
$n \geq \Omega((d/\varepsilon^2)\, e^{-m\varkappa/d})$
target samples for excess risk $\leq \varepsilon$. The optimal source allocation is $m^* = \Theta(d \log(d/\varepsilon^2)/\varkappa)$.

The exponential factor $e^{-m\varkappa/d}$ quantifies the value of source data: it decays exponentially in the source sample size m and the domain similarity $\varkappa$. For $\varkappa \to 0$ (identical domains), a single source sample completely eliminates the target sample requirement. For $\varkappa = \infty$ (completely different domains), no source data helps.

7.7 Chapter Summary

- Fano's inequality, Le Cam, and Assouad are the three principal techniques for information-theoretic lower bounds.
- Deep networks achieve the minimax rate $n^{-2s/(2s+d)}$ for compositional functions, exponentially better than shallow networks (Theorem 7.4).
- Compression implies generalisation: $I(A(S);S) \leq C$ gives gap $\leq \sqrt{(C/n)}$ (Theorem 7.5).
- CMI refines the bound to distinguish overfitting from memorisation (Theorem 7.6).
- Transfer learning requires exponentially fewer target samples when source and target are similar (Theorem 7.7).

Chapter 8

Synthesis — A Unified Information-Theoretic Framework

This chapter unifies the four frameworks developed in Chapters 1–4 (Information Bottleneck, PAC-Bayes, MDL, and MI diagnostics) into a single information-theoretic account of deep learning. We show that all four frameworks are four parametrisations of a single compression-prediction trade-off, that the optimal algorithm across all four frameworks is the Gibbs algorithm, and that the optimal prior is the Jeffreys prior weighted by the Fisher information volume.

8.1 The Four Pillars and Their Common Structure

Each of the four frameworks studied in this volume can be summarised by a triple (Complexity measure, Fit measure, Trade-off parameter):

Framework	Complexity	Fit	Trade-off	Optimal distribution
IB (Ch.1)	I(X;T)	I(T;Y)	β	Gibbs encoder p*(t|x)
PAC-Bayes (Ch.2)	D_KL(Q ∥ P)	L̂(Q)	λ	Gibbs posterior Q*_λ
MDL (Ch.3)	L_code(h)	−log p(S|h)	s	NML p_NML
MI Diag. (Ch.4)	I_σ(X;T^ℓ)	I_σ(T^ℓ;Y)	σ	Noise-reg. encoder q_σ

The common structure is: minimise (complexity) − β × (fit), where the trade-off parameter β controls the balance between compression and prediction. All four frameworks are special cases of this template.

8.2 The Fundamental Triangle

Theorem 8.1 (Information-Theoretic Triangle).
Let $Q^*_\lambda \propto P\, e^{\{-\lambda L\}}$ be the Gibbs posterior, $L_{code}(h) = -\log P(h)$ the MDL code length, and $I(A(S);S)$ the IZS mutual information. Then:
(i) $I(A(S);S) \le D_{KL}(Q^*_\lambda \| P) \le \lambda n \hat{L}(Q^*_\lambda)$ [IZS ≤ PAC-Bayes].
(ii) $D_{KL}(Q \| P) \le E_Q[L_{code}(h)] \log 2 - H(Q)$ [PAC-Bayes ≤ MDL].

(iii) At the NML optimum: SC(x^n|M) = min_{p(x̂|x)} I(X;X̂) + nL(X̂) [MDL = IB].
Together, these inequalities form a triangle: IZS ≤ PAC-Bayes ≤ MDL = IB at optimum.

The triangle inequalities have a clear interpretation: the IZS MI bound is the loosest (it only bounds the expected gap), the PAC-Bayes bound is tighter (it bounds the distribution over datasets), and the MDL bound is the most stringent (it bounds per-sequence regret). At the optimum of all three, they all equal nR*(D) — the rate-distortion function evaluated at the achieved distortion level.

8.3 The Compression-Prediction Duality

Theorem 8.2 (Compression-Prediction Duality).
The following four quantities are equivalent up to O(log n / n) terms for i.i.d. data:
1. (Rate-distortion) The minimum rate R(D) for prediction distortion ≤ D.*
2. (IB) min_{p(t|x): I(T;Y)≥H(Y)−D} I(X;T) = R(D).*
3. (MDL) SC(y^n|x^n, M) / n → R(D) at the optimal model order.*
4. (PAC-Bayes) min_{Q: E_Q[L(h)]≤D} D_KL(Q ‖ P) = nR(D) + O(log n).*
All four equal nR(D), where R*(D) is the information rate needed to predict Y from X with distortion D.*

The compression-prediction duality is the central theoretical result of this volume. It establishes that learning is compression: the amount of information that must be transmitted from training data to model is exactly the rate needed to achieve the desired prediction accuracy. This is not an approximation or an analogy — it is an exact equivalence, up to finite-sample corrections of order O(log n/n).

8.4 The Optimal Algorithm: Gibbs Sampling

Definition 8.1 (The Gibbs Algorithm).
The information-theoretically optimal learning algorithm is the Gibbs algorithm:
$A^*(S) = Q^*_\beta(h) \propto P(h) \cdot e^{-\beta n \hat{L}(h)}$,
where P is the prior, $\hat{L}(h)$ is the training loss, and $\beta > 0$ is the inverse temperature.
A(S) simultaneously: (i) minimises IZS MI, (ii) achieves the PAC-Bayes-optimal posterior (Theorem 2.4), (iii) is the Bayesian posterior — the MDL-optimal estimator (Proposition 3.2), (iv) solves the weight-space IB (Proposition 2.2).*

Remark 8.1 (SGD as Approximate Gibbs Sampling).
Stochastic gradient descent with learning rate η, batch size B, and weight decay λ approximately samples from the Gibbs distribution at effective temperature:
$T_{\text{eff}} \approx \eta \cdot n / (B \cdot \| \nabla^2 L(w^*) \|)$,
where $\| \nabla^2 L(w^*) \|$ is the spectral norm of the Hessian at the minimum. The effective temperature is controlled by the ratio η/B: small η or large B drives $T_{\text{eff}} \to 0$ (converging to the MAP solution), while large η or small B maintains

$T_{eff} > 0$ (exploring the posterior). This is the information-theoretic explanation for the generalisation advantage of small-batch SGD: it maintains a higher effective temperature, exploring flatter minima.

8.5 The Optimal Prior: Jeffreys

Proposition 8.1 (Jeffreys Prior is Optimal).
The prior P^* that minimises the PAC-Bayes bound, MDL stochastic complexity, and IB complexity simultaneously, in the limit $n \to \infty$, is the Jeffreys prior:
$P^*(w) \propto \sqrt{\det I_F(w)}$, *where $I_F(w)$ is the Fisher information matrix of the network output.*
The Jeffreys prior concentrates on flat regions of the loss landscape (where $\det I_F$ is small), precisely the regions associated with good generalisation.

The Jeffreys prior has several remarkable properties that explain its optimality:

- Reparametrisation invariance: the Jeffreys prior assigns the same probability to the same physical hypothesis regardless of the parametrisation, eliminating arbitrary coordinate dependencies.
- Minimax MDL: under the Jeffreys prior, the stochastic complexity $SC(x^n|M)$ achieves the minimum worst-case redundancy (Theorem 3.2).
- Flat minima preference: $\det I_F(w)$ is small at flat minima of the loss landscape (where the Hessian has small eigenvalues), so the Jeffreys prior assigns high probability to flat minima. This is the information-theoretic justification for sharpness-aware minimisation.

8.6 Ten Key Theorems of Volume III

T1 (IB Self-Consistency).
The optimal IB encoder $p^*(t|x) \propto p(t) \exp(-\beta\, D_{KL}(p(y|x) \,/\!/\, p(y|t)))$ satisfies fixed-point equations whose iterative solution converges to the IB curve C_{IB}.

T2 (PAC-Bayes Bound).
With probability $\geq 1-\delta$: $L(Q) \leq \hat{L}(Q) + \sqrt{[(D_{KL}(Q \,/\!/\, P) + \log 2\sqrt{n}/\delta)/(2n)]}$, uniformly over all Q.

T3 (Flat Minima and Generalisation).
$L(w^*) \leq \text{Sharp}_\rho(w^*) + O(\sqrt{[\,/\!/\, w^* \,/\!/\,^2/\rho^2 n]})$: sharpness at scale ρ controls the generalisation bound via the perturbation posterior KL divergence.

T4 (MDL Minimax Optimality).
The NML code achieves minimax regret $\log C_n(M)$ with complexity $\log C_n(M) = (d/2)\log n + \log \text{Jeffreys-volume} + o(1)$.

T5 (ELBO = MDL for Latent Models).

The negative ELBO $-L_{ELBO} = E_q[-\log p(x|z)] + D_{KL}(q(z|x) \,/\!/\, p(z))$ is the MDL two-part code for latent variable models; VAE training = compression.

T6 (MI and Intrinsic Dimension).

$I_\sigma(X;T^\ell) = (d_\ell/2)\log(1/\sigma^2) + O(1)$ as $\sigma \to 0$, where $d_\ell = \dim_H(\mathrm{supp}\, T^\ell)$: MI slope estimates intrinsic dimension.

T7 (β-VAE Rate-Distortion).

β-VAE traces the rate-distortion curve of p(x); water-filling identifies informationally relevant principal components of the data covariance.

T8 (Deep Network Minimax Rate).

Minimax rate for compositional functions is $\Theta(n^{-2s/(2s+d)})$, achieved by depth-matched ReLU networks. Overparameterisation does not worsen this rate.

T9 (Compression Implies Generalisation).

$E[L(A) - \hat{L}(A)] \leq \sqrt{(2\sigma^2 I(A(S);S)/n)}$: algorithms that transmit $\leq C$ bits from data to model have generalisation gap $\leq \sqrt{(2\sigma^2 C/n)}$.

T10 (Gibbs Algorithm Optimality).

$A^*(S) \propto P\, e^{-\beta n \hat{L}}$ simultaneously minimises IZS MI, PAC-Bayes KL, MDL stochastic complexity, and IB functional — the unique information-theoretically optimal learning algorithm.

8.7 Open Problems

Problem 8.1.

Tight CMI bounds for transformer LLMs with unbounded cross-entropy loss: current bounds require sub-Gaussian or bounded losses.

Problem 8.2.

Efficient computation of the MDL stochastic complexity for networks with $p > 10^7$ parameters: the Fisher information matrix is intractable at this scale.

Problem 8.3.

Information-theoretic theory of in-context learning: how does attending to k in-context examples change the effective posterior over hypotheses?

Problem 8.4.

Wasserstein rate-distortion theory for non-decomposable distortion measures.

Problem 8.5.

Information-theoretic lower bounds on the sample complexity of fine-tuning foundation models.

8.8 Closing Remarks

The central message of this volume is structural. Deep learning succeeds not in spite of information-theoretic principles but because of them: deep networks are matched to the

compositional structure of natural data (Theorem 7.4), SGD is an approximate Gibbs sampler (Remark 8.1), overparameterisation achieves low complexity in the PAC-Bayes sense (Theorem 7.5), and the intrinsic dimension of data is far smaller than the ambient dimension (Proposition 5.1).

Each of these facts is an information-theoretic fact. The frameworks developed in this volume — IB, PAC-Bayes, MDL, and MI diagnostics — are not four separate theories. They are four windows onto the same mathematical reality: the fundamental trade-off between compression and prediction that underlies all of statistical learning. The Gibbs algorithm (Definition 8.1) and the Jeffreys prior (Proposition 8.1) are the information-theoretically canonical answers to the questions 'which algorithm should we use?' and 'which prior should we choose?'.

The mathematics of information is the language of learning. Volume IV will continue this programme by developing the computational complexity theory of deep learning, asking not just what is information-theoretically achievable but what is computationally achievable within that envelope.

Chapter 9

Information Theory and Differential Privacy

Differential privacy (DP) provides a rigorous mathematical guarantee that the output of a learning algorithm reveals negligible information about any individual training data point. Originally developed in the cryptography literature (Dwork et al. 2006), DP has become the dominant privacy framework for machine learning, with deployments at Google, Apple, Meta, and the US Census Bureau. This chapter develops the information-theoretic foundations of DP and establishes its connection to the rate-distortion and MDL frameworks of the preceding chapters.

9.1 Differential Privacy: Definitions

Definition 9.1 (Differential Privacy).
A randomised mechanism $M : D^n \to O$ satisfies (ε,δ)-differential privacy if for all adjacent datasets S, S' (differing in one element) and all measurable $O \subseteq O$:
$P(M(S) \in O) \leq e^{\varepsilon} \cdot P(M(S') \in O) + \delta.$
When $\delta = 0$: pure ε-DP. When $\delta > 0$: approximate (ε,δ)-DP.
Interpretation: ε bounds the log-likelihood ratio of any outcome under $M(S)$ vs. $M(S')$. An adversary observing $M(S)$ gains at most e^{ε} advantage in guessing whether any individual is in S.

The ε parameter has a natural information-theoretic interpretation: it is the maximum log-likelihood ratio, which bounds the KL divergence $D_{KL}(M(S) \,\|\, M(S')) \leq \varepsilon(e^{\varepsilon} - 1) \leq \varepsilon^2$ (for $\varepsilon \leq 1$). Smaller ε means the mechanism reveals less information about individual data points.

Key composition properties of DP:

- Sequential composition: if M_1 is ε_1-DP and M_2 is ε_2-DP (applied to the output of M_1), then (M_1, M_2) is $(\varepsilon_1+\varepsilon_2)$-DP. Privacy budgets accumulate linearly.
- Parallel composition: if M_1 operates on disjoint subsets of the data, the combined mechanism is $\max(\varepsilon_1,\varepsilon_2)$-DP.
- Post-processing: any function of a DP mechanism output is DP with the same (ε,δ). Privacy cannot be weakened by post-processing.

9.2 Rényi Differential Privacy

Definition 9.2 (Rényi DP).
A mechanism M satisfies (a,ε)-Rényi DP (RDP) if for all adjacent S, S':
$D_a(M(S) \,\|\, M(S')) \leq \varepsilon,$ *where* $D_a(P \,\|\, Q) = (1/(a-1)) \log E_Q[(dP/dQ)^a].$
RDP provides tighter composition bounds: k applications of (a,ε_0)-RDP give $(a, k\varepsilon_0)$-RDP.

Theorem 9.1 (Gaussian Mechanism is RDP).

The Gaussian mechanism $M(S) = f(S) + N(0, \sigma^2 I)$ with sensitivity $\Delta_f = \max_{S \sim S'} \| f(S) - f(S') \|_2$ satisfies:
$(a, a\Delta_f^2/(2\sigma^2))$-RDP for all $a > 1$.
Converting to (ε,δ)-DP: for any $\delta > 0$, M satisfies $(a\Delta_f^2/(2\sigma^2) + \log(1/\delta)/(a-1), \delta)$-DP.

9.3 DP-SGD: Private Neural Network Training

Definition 9.3 (DP-SGD, Abadi et al. 2016).
At each iteration t with mini-batch B_t:
1. Compute per-example gradients $g_t(x_i) = \nabla_w \ell(f_w(x_i), y_i)$ for $x_i \in B_t$.
2. Clip: $\hat{g}_t(x_i) = g_t(x_i) / \max(1, \| g_t(x_i) \|_2 / C)$.
3. Aggregate and noise: $\tilde{g}_t = (1/|B_t|)(\Sigma_i \hat{g}_t(x_i) + N(0, \sigma^2 C^2 I))$.
4. Update: $w_{t+1} = w_t - \eta \tilde{g}_t$.

Theorem 9.2 (Privacy of DP-SGD).
DP-SGD with clipping norm C, noise σ, sampling rate $q = |B|/n$, and T iterations satisfies $(a, qa/(2\sigma^2))$-RDP per step and approximately:
$\varepsilon \approx q\sqrt{(T \log(1/\delta))} / \sigma$ (ε,δ)-DP for large T.
The privacy cost grows as $\sqrt{T}$ with iterations — each additional training step costs $\sqrt{1/T}$ privacy budget.

The privacy-utility trade-off in DP-SGD: increasing σ reduces ε (better privacy) but also reduces the signal-to-noise ratio of the gradient, slowing convergence. The optimal σ balances ε against convergence rate. Empirically, DP models trained with $\varepsilon \approx 1$–10 on standard benchmarks (MNIST, CIFAR-10) achieve accuracy within 5–10% of non-private models, with larger accuracy gaps for smaller datasets or more complex models.

9.4 The Privacy-Utility Trade-off as Rate-Distortion

The privacy-utility trade-off has the same mathematical structure as the rate-distortion problem. Define:

- Rate = privacy loss ε: how much information the output reveals about individual data points.
- Distortion = utility loss: the degradation in model accuracy due to privacy constraints.

Theorem 9.3 (Privacy-Utility as Rate-Distortion).
The privacy-utility curve $\{(\varepsilon^*, U^*) : M$ achieves privacy ε and utility $U\}$ is equivalent to the rate-distortion curve of the learning problem under the Rényi divergence as rate measure:
$C_{PU} = \{(D_\infty(M(S) \| M(S')), I(M(S);S)) : M$ ranges over all (ε,δ)-DP mechanisms$\}$.

The optimal mechanism for a given ε (achieving minimum distortion at rate ε) is the Gibbs mechanism $M^*(S) \propto P(h)\, e^{-\varepsilon n \hat{L}(h)}$, which is the DP analogue of the IB-optimal encoder.

This theorem reveals that the privacy-utility trade-off is a rate-distortion problem: privacy loss plays the role of rate (bits transmitted from data to output), and utility plays the role of negative distortion. The fundamental limit is set by the privacy-utility rate-distortion curve, below which no mechanism can achieve both strong privacy and high utility simultaneously.

9.5 Lower Bounds on Private Learning

Theorem 9.4 (Private PAC Learning Requires More Samples).
Any (ε,δ)-DP PAC learner for H with VC dimension d achieving excess risk $\leq 1/4$ with probability $\geq 2/3$ requires:
$n = \Omega(d/(\varepsilon \log(1/\delta)))$
samples — a factor of $\Omega(d/(\varepsilon \log(1/\delta)))$ more than non-private learning.
Proof: DP constrains $I(M(S);S) \leq O(n\varepsilon^2)$ by the Gaussian mechanism analysis. Fano's inequality then requires $n\varepsilon^2 \geq \Omega(d)$ to distinguish d hypotheses, giving $n = \Omega(d/\varepsilon^2)$.

The lower bound shows a fundamental incompatibility between privacy and sample efficiency: to learn with ε-privacy, you need $\Theta(1/\varepsilon^2)$ more data than without privacy. This is the information-theoretic price of privacy: each bit of information the mechanism reveals about individual data points reduces the learning capacity by a corresponding factor.

9.6 Chapter Summary

- DP bounds the log-likelihood ratio of mechanism outputs under adjacent datasets (Definition 9.1).
- Rényi DP provides tight composition bounds (Definition 9.2, Theorem 9.1).
- DP-SGD achieves (ε,δ)-DP with $\varepsilon \approx q\sqrt{T}/\sigma$ (Theorem 9.2).
- Privacy-utility trade-off = rate-distortion problem (Theorem 9.3).
- Private learning requires $\Omega(d/\varepsilon^2)$ samples vs. $\Omega(d)$ non-private (Theorem 9.4).

Chapter 10

Information Geometry of Neural Network Training

Information geometry studies the Riemannian geometry of statistical manifolds, equipping the space of probability distributions with the Fisher metric and associated connections. For neural networks, this geometry provides a principled account of training dynamics, the connection between Adam and natural gradient, and the relationship between curvature and generalisation. This chapter develops the key concepts and applies them to the analysis of deep learning.

10.1 The Fisher Metric and the Statistical Manifold

A parametric family of distributions $\{p(x|w) : w \in R^p\}$ forms a statistical manifold M. The Fisher information matrix:

$$G(w)_{ij} = E_{p(x|w)}[\partial_i \log p(x|w) \cdot \partial_j \log p(x|w)] = I_F(w)_{ij}$$

is the natural Riemannian metric on M. The geodesic distance between two distributions on M in the Fisher metric is the Fisher-Rao distance, which is reparametrisation-invariant (unlike Euclidean distance in parameter space).

Definition 10.1 (Natural Gradient).
The natural gradient of a loss $L(w)$ on the statistical manifold is:
$\tilde{\nabla}_w L(w) := I_F(w)^{-1} \nabla_w L(w)$.
Natural gradient descent $w \leftarrow w - \eta \tilde{\nabla}_w L(w)$ is steepest descent in the Fisher-Rao metric.
Key property: natural gradient descent is reparametrisation-invariant — the update direction is the same regardless of how the model is parametrised.

The reparametrisation invariance of natural gradient is its fundamental advantage over ordinary gradient descent. Consider a network with two equivalent parametrisations: the standard weight parametrisation w and a reparametrised version $\varphi = f(w)$. Ordinary gradient descent in w and in φ give different update directions (related by the Jacobian of f). Natural gradient descent in w and in φ give the same update direction (in terms of the induced distribution over outputs), because the Fisher metric absorbs the Jacobian.

10.2 Adam as Diagonal Natural Gradient

Proposition 10.1 (Adam Approximates Natural Gradient).
Adam's update at step t: $w_{t+1} = w_t - \eta \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon)$ approximates natural gradient when:
$I_F(w)_{ii} \approx E[g^2_i] \approx v_{t,i} / (1 - \beta^{2t})$,
i.e., when the Fisher matrix is approximately diagonal. This holds for models with independent parameters (diagonal exponential families) but not for models with correlated parameters (attention layers in transformers).

Adam's convergence advantage over SGD is explained by this natural-gradient approximation: it adapts the learning rate to the curvature of the loss landscape, taking larger steps in flat directions and smaller steps in sharp directions.

Exact natural gradient methods use block-diagonal (K-FAC) or Kronecker-factored approximations to I_F, which capture correlations within layers while remaining computationally tractable. K-FAC achieves faster convergence than Adam for large language models but requires approximately 3× more memory for the second-order information.

10.3 Geodesics, Duality, and the EM Algorithm

Definition 10.2 (The α-Connections).
Information geometry equips M with a one-parameter family of connections (a-connections) for a ∈ R. The cases a = +1 (e-connection, exponential) and a = −1 (m-connection, mixture) are dual with respect to the Fisher metric.
Under the e-connection, exponential families are flat: their geodesics are straight lines in the natural parameter space η.
Under the m-connection, mixture families are flat: their geodesics are straight lines in the mean parameter space μ.

Theorem 10.1 (Pythagorean Theorem in Information Geometry).
Let M_e be e-flat and M_m be m-flat, with p their intersection. For p ∈ M_e and q ∈ M_m:*
D_KL(q ‖ p) = D_KL(q ‖ p) + D_KL(p* ‖ p).*
The EM algorithm implements this decomposition: the E-step is m-projection (onto M_m, the family of posterior expectations) and the M-step is e-projection (onto M_e, the model family). Both steps decrease D_KL(data ‖ model).

10.4 Loss Landscape Curvature and Generalisation

Proposition 10.2 (Fisher = Hessian at Minimum).
At a minimum w of the cross-entropy loss L(w) = −E_{p(x,y)}[log p_w(y | x)]:*
∇²_w L(w) = I_F(w*).*
The Hessian of the loss at a minimum equals the Fisher information matrix. Flat minima (small Hessian eigenvalues) correspond to directions where the task-relevant data provides little information about the weights.

Theorem 10.2 (Effective Dimensionality and PAC-Bayes).
The effective dimensionality of the model at minimum w is:*
d_eff(w) = tr(I_F(w*)) / λ_max(I_F(w*)) ≤ p.*
The PAC-Bayes bound (Theorem 2.5) with ρ² = 1/λ_max(I_F(w)) is bounded by O(d_eff(w*)/n).*

Good generalisation requires $d_{\text{eff}}(w^*) \ll p$ — the model must be 'effectively low-dimensional' at the minimum, i.e., most weight directions must be unconstrained by the data.

The effective dimensionality $d_{\text{eff}} = \text{tr}(I_F)/\lambda_{\max}$ equals the number of directions in weight space that are 'strongly constrained' by the data (eigenvalues of I_F comparable to $\lambda_{\max}$). For overparameterised networks where $d_{\text{eff}} \ll p$, the PAC-Bayes bound is non-vacuous even though $p \gg n$. This is the geometric explanation for the generalisation of overparameterised networks: they find solutions that are effectively low-dimensional.

10.5 Chapter Summary

- The Fisher metric makes the space of probability distributions a Riemannian manifold (Definition 10.1).
- Natural gradient descent is reparametrisation-invariant; Adam is its diagonal approximation (Proposition 10.1).
- The Pythagorean theorem in information geometry underlies the EM algorithm and variational inference (Theorem 10.1).
- Fisher matrix = Hessian at loss minimum (Proposition 10.2).
- Effective dimensionality d_{eff} controls the PAC-Bayes bound; good generalisation requires $d_{\text{eff}} \ll p$ (Theorem 10.2).

Chapter 11

Federated Learning and Communication-Constrained Inference

Federated learning (FL) trains models across distributed devices without centralising raw data. Each of K clients holds a local dataset S_k and communicates only model updates to a central server. The key constraint is communication: each client can send at most B bits per round. This chapter develops the information-theoretic foundations of FL, establishing connections to rate-distortion theory (for gradient compression), PAC-Bayes theory (for distributed generalisation bounds), and differential privacy (for private federated learning).

11.1 The Federated Setting

Definition 11.1 (Federated Learning Protocol).
At each communication round t:
1. Server broadcasts global model w_t (B_server bits per client).
2. Each client k computes local update ⊿_k(w_t, S_k) and transmits Q_B(⊿_k) (≤ B bits).
3. Server aggregates: w_{t+1} = w_t + (1/K) Σ_k Q_B(⊿_k).
Total communication per round: K·B bits (client → server) + K·B_server bits (server → client).

The key challenge is that clients have heterogeneous local distributions D_k ≠ D (data heterogeneity or 'non-IID data'), which means that local updates Δ_k point in different directions. Aggregating these updates produces an estimate of the global gradient ∇_w E_{D_k}[L(w)] that may differ significantly from the desired ∇_w L(w) on the global distribution D = (1/K)Σ_k D_k.

11.2 Gradient Compression as Rate-Distortion

Theorem 11.1 (Optimal Gradient Compression Rate).
For gradient g ~ N(0, σ²I_p) and squared-error distortion, the minimum bits per coordinate to achieve distortion E[‖ g − Q_B(g) ‖ ²/p] ≤ D is:
b(D) = (1/2) max{0, log(σ²/D)} bits/coordinate,*
achieved by water-filling. Total bits: B = p·b*(D) = (p/2) log(σ²/D).*
Uniform b-bit quantisation achieves B = pb bits with distortion D = σ²·(2^{-2b}/3) (quantisation noise), achieving the water-filling rate up to a 3/π² ≈ 0.3 bit gap.

The gradient compression problem is a rate-distortion problem with:

- Source: the local gradient g_k ~ N(0, σ²I_p) (under Gaussian approximation).
- Channel: the communication link from client k to the server (B bits).
- Distortion: the squared-error loss ‖ g_k − Q_B(g_k) ‖ ².

Sparsification (top-k selection): transmit only the k largest-magnitude gradient coordinates, with $k \leq p/\log p$ (to send indices efficiently). The communication cost is $k(\log p + 32)$ bits, and the distortion is the sum of squared magnitudes of the discarded coordinates.

11.3 Federated PAC-Bayes

Theorem 11.2 (Federated PAC-Bayes Bound).
With K clients each holding n_k samples, local posteriors Q_k, and aggregate posterior $Q = (1/K)\Sigma_k Q_k$, for any prior P, with probability $\geq 1 - \delta$:
$L(Q) \leq (1/K)\Sigma_k \hat{L}_{S_k}(Q) + \sqrt{[((1/K)\Sigma_k D_{KL}(Q_k \| P) + \log(2K\sqrt{n_{min}}/\delta))/(2n_{min})]}$,
where $n_{min} = \min_k n_k$.
The bound improves when all clients have many samples (n_{min} large) and when local posteriors are close to the prior (aggregated KL is small).

The federated PAC-Bayes bound reveals the information-theoretic structure of federated generalisation:

- Aggregated KL: $(1/K)\Sigma_k D_{KL}(Q_k \| P)$ measures the average deviation of local posteriors from the prior. Large KL means clients have made large updates from the prior, capturing a lot of local information — which is good for local performance but may hurt global generalisation.
- Sample size: n_{min} is the minimum client dataset size. The bound degrades as the weakest client gets fewer samples, even if other clients have many samples — federated learning is bottlenecked by the least data-rich client.

11.4 Private Federated Learning

Combining DP-SGD with federated learning gives differentially private federated learning (DP-FL). The privacy analysis must account for:

- Local privacy: each client's update must be private with respect to its local dataset.
- Central privacy: the aggregate model must be private with respect to any individual's data.

Theorem 11.3 (DP-FL Privacy Guarantee).
In federated DP-FL with K clients, local noise σ_k, clipping norm C, sampling rate q, and T rounds:
(i) Local DP: each client's update $\perp_k$ satisfies $(\varepsilon_{loc}, \delta_{loc})$-DP for $\varepsilon_{loc} \approx q\sqrt{T}/\sigma_k$.
(ii) Central DP: the aggregate satisfies $(\varepsilon_{central}, \delta_{central})$-DP for $\varepsilon_{central} \approx q\sqrt{T}/(\sqrt{K}\sigma_k)$.
The central privacy guarantee is K times stronger than local: with K clients each adding noise σ_k, the aggregate noise level is $\sigma_k\sqrt{K}$, amplifying privacy by $\sqrt{K}$.

The √K amplification of privacy in federated settings is a direct consequence of the central limit theorem: summing K i.i.d. noise terms $N(0, \sigma^2_k)$ gives $N(0, K\sigma^2_k)$, which is √K times larger than the individual noise. This allows federated settings to achieve stronger central privacy guarantees than individual clients, a key advantage of federated architectures.

11.5 Communication Complexity Lower Bounds

Theorem 11.4 (Communication Lower Bound for Federated Learning).
In the federated setting with K clients, n total training samples (n/K per client), and VC dimension d, the minimum total bits to achieve generalisation gap ≤ ε with probability ≥ 1 − δ is:
$B_total = \Omega(d \cdot n \cdot \log(1/\varepsilon))$,
independent of K. The communication cannot be distributed to reduce the total below this threshold.

The lower bound is remarkable: the total communication required to match centralised learning does not decrease with the number of clients K. This implies that federated learning cannot be 'free' — each client must contribute Ω(d log(1/ε)) bits to the central server, regardless of how many clients participate.

However, the communication can be distributed optimally across clients: each client sends Ω(d log(1/ε) / K) bits per round, and T = K rounds are sufficient. With T rounds and K clients:

$$\text{Total bits} = K \cdot T \cdot (d \log(1/\varepsilon)/K) = T \cdot d \log(1/\varepsilon) = \Omega(d \log(1/\varepsilon)),$$

matching the lower bound. Federated learning with communication compression achieves the lower bound up to constants, making it information-theoretically efficient.

11.6 Chapter Summary

- FL trains models across K clients communicating ≤ B bits/round each (Definition 11.1).
- Gradient compression is a rate-distortion problem with optimal rate $b^*(D) = (1/2) \log(\sigma^2/D)$ bits/coordinate (Theorem 11.1).
- Federated PAC-Bayes bound is controlled by the average KL of local posteriors and the minimum client dataset size (Theorem 11.2).
- DP-FL achieves √K privacy amplification in the central setting (Theorem 11.3).
- Total communication lower bound is Ω(d·n·log(1/ε)), independent of K (Theorem 11.4).

3.4b Practical Computation of Stochastic Complexity

The stochastic complexity SC(x^n|M) = −log p(x^n|θ̂(x^n)) + log C_n(M) requires computing the parametric complexity C_n(M) = Σ_{z^n} p(z^n|θ̂(z^n)), which is a sum over all possible data sequences. For most model families, C_n(M) is analytically intractable and must be approximated.

For parametric families satisfying regularity conditions, Theorem 3.2 gives the asymptotic expansion log C_n(M) ≈ (d/2) log(n/2π) + log ∫_Θ √det I(θ) dθ. The Fisher information volume ∫_Θ √det I(θ) dθ can be computed analytically for exponential families:

- Gaussian family N(μ, σ²) with both parameters unknown (d=2): ∫ √det I(μ,σ) dμ dσ = ∫ (1/σ²) dμ dσ = ∞ (improper). The Jeffreys prior p(μ,σ) ∝ 1/σ² is improper but leads to a proper normalised maximum likelihood distribution.
- Bernoulli(θ) family (d=1): ∫₀¹ √I(θ) dθ = ∫₀¹ 1/√(θ(1-θ)) dθ = π. The parametric complexity is log C_n(Bernoulli) ≈ (1/2) log(n/2π) + log π ≈ (1/2) log n + 0.57.
- Multinomial with k categories (d=k-1): log C_n(Multinomial) ≈ ((k-1)/2) log(n/2π) + log Vol(simplex under Jeffreys metric).

For neural network model classes, the Fisher information matrix is a p×p matrix (p = number of parameters), and computing det(I(w)) requires O(p³) operations — completely intractable for p > 10⁴. In practice, approximate methods are used:

- Diagonal Fisher approximation: replace I(w) by its diagonal, computing det I ≈ Π_i I_ii. This ignores all off-diagonal correlations.
- K-FAC (Kronecker-Factored Approximate Curvature): approximate I(w) by a Kronecker product of smaller matrices, one per layer. This captures within-layer correlations while remaining computationally tractable.
- Empirical Fisher: replace the expectation E[∇ log p · (∇ log p)^T] by its sample average over the training set, giving a low-rank approximation to the Fisher matrix.

3.7b MDL and Network Pruning

Network pruning — removing weights or neurons from a trained network to reduce its size and computational cost — has a natural MDL interpretation. The MDL objective for a network with weight vector w is:

$$MDL(w) = L_1(w) + L_2(S\,|\,w) = b \cdot p + n \cdot L(w)/\log 2,$$

where b is the precision (bits per weight) and p is the number of non-zero weights. Pruning reduces p (and hence L_1) at the cost of increasing L_2 (the model fits the data less well with fewer parameters). The optimal pruning decision for weight w_i is: remove w_i if the increase in L_2 from removing it is less than the decrease in L_1 from removing it.

For a quantised network with b bits per weight, the decrease in L_1 from removing one weight is b bits. The increase in L_2 from removing weight w_i is approximately $(1/2) \cdot (w_i/\sigma_i)^2$ (where σ_i is the posterior standard deviation of w_i estimated from the Hessian of the loss). The optimal MDL pruning criterion is therefore:

$$\text{Prune } w_i \text{ if } \quad w_i^2/\sigma_i^2 < 2b \log 2.$$

This is the Optimal Brain Damage (OBD) criterion of LeCun et al. (1990) and the Optimal Brain Surgeon (OBS) of Hassibi and Stork (1993), now derived from MDL first principles. The factor 2b log 2 is the MDL threshold: weights whose saliency w_i^2/σ_i^2 falls below this threshold should be removed.

Magnitude-based pruning (removing weights with $|w_i|$ below a threshold) is a cruder approximation: it assumes σ_i is constant across all weights, which is not true in general (different weights have different posterior uncertainties depending on their contribution to the loss). MDL-based pruning (using the full saliency metric w_i^2/σ_i^2) consistently outperforms magnitude-based pruning by 5–15% in accuracy at the same compression ratio.

4.3b MI Estimation in Practice: A Comparative Study

Choosing the right MI estimator for a given application requires understanding the trade-offs between consistency, computational cost, and variance. We summarise the key properties of the main estimators.

MINE (Mutual Information Neural Estimation): consistent for any distribution; computationally expensive (requires training a neural critic); high variance for small batch sizes (bias $O(1/n)$); scales to arbitrary high dimensions; requires specification of critic architecture.

KSG (Kraskov-Stögbauer-Grassberger): consistent for smooth distributions; $O(n \log n)$ computational cost; bias $O(k/n)$ with variance $O(1/k)$; works well up to dimension $d \approx 20$; no hyperparameter beyond k (number of neighbours); fails for high-dimensional data.

NWJ (Nguyen-Wainwright-Jordan): consistent; lower variance than MINE but higher bias; no log in the denominator avoids Jensen's inequality bias; scales to high dimensions; requires critic network.

InfoNCE (van den Oord et al.): biased downward by $O(\log K)$ where K is the number of negative samples; computationally efficient ($O(K^2)$ per batch); widely used in contrastive learning; provides a lower bound on $\min(I(X;T), \log K)$.

A practical recommendation: for activations with $d \leq 10$, use KSG with $k = 5$–10; for $d \in [10, 50]$, use KSG with dimensionality reduction (PCA to 10 dimensions); for $d > 50$, use MINE or NWJ with a 2-layer MLP critic; for contrastive learning applications, use InfoNCE.

4.6b Information-Theoretic Measures of Representation Quality

Beyond MI, several derived information-theoretic quantities provide useful diagnostics of representation quality.

Conditional entropy $H(Y|T^\ell)$: the uncertainty in Y given the representation T^ℓ. A good representation has low $H(Y|T^\ell)$, meaning T^ℓ predicts Y well. $H(Y|T^\ell) = H(Y) - I(T^\ell;Y)$ is directly computable if $I(T^\ell;Y)$ is estimated.

Redundancy: $I(T^\ell;X) - I(T^\ell;Y) = I(T^\ell; X|Y)$. This measures the information in T^ℓ about X that is not useful for predicting Y — the 'wasted information'. A representation with low redundancy retains only the information about X that is relevant for Y, consistent with the IB optimal solution.

Synergy: $I(X;Y|T^\ell) - I(X;Y) + I(T^\ell;Y)$. This measures the additional information about Y that is present in X but not captured by T^ℓ — the 'missed information'. A representation with low synergy captures all the available information about Y from X.

The Partial Information Decomposition (PID, Williams and Beer 2010) provides a finer decomposition of $I(X;Y;T^\ell)$ into unique, redundant, and synergistic components, giving a complete picture of the information flow through layer ℓ. However, PID is computationally expensive and requires additional assumptions to be uniquely defined for more than two sources.

5.3b The Manifold Hypothesis: Evidence and Caveats

The manifold hypothesis states that high-dimensional data (images, audio, text) lies near a low-dimensional manifold embedded in the ambient space. This hypothesis motivates dimensionality reduction, representation learning, and the expectation that deep networks can generalise despite the curse of dimensionality.

Evidence for the manifold hypothesis:

- Intrinsic dimension estimates for natural image datasets: ImageNet images ($224\times224\times3 = 150{,}528$ dimensions) have estimated intrinsic dimension 40–100 (depending on the estimation method). MNIST digits ($28\times28 = 784$ dimensions) have estimated intrinsic dimension 10–15.
- Generative models: modern image generators (GANs, VAEs, diffusion models) operate in a latent space of dimension 100–1000 and produce photorealistic images, suggesting that natural images can be accurately described by a low-dimensional latent representation.
- Transfer learning: features learned on large datasets transfer to new tasks with few examples, suggesting that the useful features are a low-dimensional subset of the input space.

Caveats and limitations:

- The manifold hypothesis is approximate: real data does not lie exactly on a smooth manifold but near one (with observation noise). The effective dimension depends on the noise level and the precision of the intrinsic dimension estimator.
- Different regions of the data space may have different intrinsic dimensions: rare categories or unusual images may lie on higher-dimensional submanifolds than common categories.
- The intrinsic dimension depends on the task: for classification, the relevant intrinsic dimension is $\log_2 C$ (one bit per binary classification decision); for generation, it is much higher (the full variation in the data).

The Hypothesis 5.1 (Rectifiable Representation Hypothesis) provides a mathematical formalisation that avoids these caveats: it requires only that the representation support be approximately d-rectifiable, not that it lie on a smooth manifold. Rectifiable sets can have singularities, boundaries, and variable local dimension, making the hypothesis more realistic.

5.5b Wasserstein Distance and Generative Model Evaluation

The Fréchet Inception Distance (FID) is the de facto standard for evaluating generative models. It is defined as the W_2 distance between the distributions of Inception-v3 features for real and generated images:

$$FID = W_2^2(N(\mu_{real}, \Sigma_{real}), N(\mu_{gen}, \Sigma_{gen})) = \| \mu_{real} - \mu_{gen} \|^2 + tr(\Sigma_{real} + \Sigma_{gen} - 2(\Sigma_{real}^{1/2} \Sigma_{gen} \Sigma_{real}^{1/2})^{1/2}).$$

The Gaussian approximation in FID is a simplification (real and generated feature distributions are not Gaussian), but it provides a computationally tractable proxy for the true W_2 distance between the feature distributions. FID correlates well with human judgements of image quality, making it a practical evaluation metric.

Information-theoretic interpretation of FID: the FID measures the statistical distance between the real and generated distributions in the feature space of Inception-v3. Low FID means the generative model produces features that are statistically similar to real features — the model has learned to capture the distribution of real features. High FID means the generative model's features differ significantly from real features, indicating either mode collapse (the model generates a limited variety of images) or mode dropping (the model fails to generate certain types of images).

The optimal transport interpretation: the W_2 distance between Gaussians has an explicit formula (as given above), and the optimal coupling is achieved by the linear map $T^* = \Sigma_{gen}^{1/2}(\Sigma_{gen}^{1/2} \Sigma_{real} \Sigma_{gen}^{1/2})^{-1/2} \Sigma_{gen}^{1/2}$ (the Bures metric). This coupling provides the 'ideal interpolation' between the real and generated distributions, which is useful for understanding how to improve the generative model.

6.3b The Rate-Distortion-Perception Triple

Blau and Michaeli (2018) showed that the trade-off between rate, distortion, and perceptual quality forms a 3-dimensional surface — the Rate-Distortion-Perception (RDP) triple. At any fixed rate R, the achievable (distortion, perception) pairs form a perception-distortion curve, and as R increases, this curve shifts toward lower distortion and better perception.

The RDP surface has three regimes:

- High rate ($R \to H(X)$): distortion $\to 0$ and perception $\to 0$ simultaneously (lossless compression achieves both). Both MSE and perceptual quality can be achieved at high rates.

- Medium rate: distortion and perception trade off. Minimising MSE (high distortion) gives poor perception; maximising perceptual quality gives high MSE. The Pareto frontier is the perception-distortion curve at rate R.
- Low rate ($R \to 0$): distortion $\to$ Dmax and perception $\to \Delta$max. Both MSE and perceptual quality degrade at low rates, but they degrade differently: MSE-optimal reconstruction at low rates produces blurry averages, while perception-optimal reconstruction produces sharp but potentially hallucinated details.

This framework provides a principled basis for designing image compression algorithms that balance perceptual quality against MSE, depending on the application (medical imaging prioritises MSE; social media sharing prioritises perceptual quality).

6.5b Mode Collapse in GANs: An Information-Theoretic Analysis

Mode collapse — the phenomenon where a GAN generator produces a limited variety of outputs, ignoring large portions of the data distribution — is one of the most common failure modes of GAN training. The information-theoretic analysis reveals why mode collapse occurs and how to prevent it.

In the rate-distortion perspective, a GAN with a k-dimensional latent code z can represent at most k 'effective dimensions' of variation in the generated distribution G#p(z). If the data distribution p(x) has intrinsic dimension $d > k$, the generator cannot cover the entire data manifold — it can only approximate d of the data directions, and the remaining $d - k$ directions are ignored (collapsed).

Formally, mode collapse corresponds to the generator G reducing the intrinsic dimension of its output distribution: $\dim_H(G\#p(z)) < \dim_H(p(x))$. This is precisely the information-theoretic compression that the IB seeks to avoid: the generator compresses the data representation more than necessary, losing information about the data distribution.

The information-theoretic remedy is to add an explicit mutual information term $I(z; G(z))$ to the GAN objective, encouraging the generator to use all dimensions of z to produce diverse outputs. This is the motivation for InfoGAN (Chen et al. 2016), which maximises $I(z; G(z))$ while training the GAN, and for mutual information regularised GANs more generally.

7.4b Double Descent and the Information-Theoretic Perspective

The double descent phenomenon (Belkin et al. 2019) refers to the non-monotonic test error curve observed in modern overparameterised models: as the model size (or training time) increases, the test error first decreases (classical regime), then increases (overfitting regime near the interpolation threshold), then decreases again (modern overparameterised regime). The information-theoretic analysis provides a unified explanation.

Classical regime ($p < n$): the model is underparameterised, and adding parameters reduces bias more than it increases variance. The test error decreases as p increases. The minimax rate $n^{-2s/(2s+d)}$ is achieved at the optimal model size $p^* = n^{d/(2s+d)}$.

Interpolation threshold ($p \approx n$): the model has just enough parameters to interpolate the training data. The minimum-norm interpolant has high norm $\|w^*\|$, which gives a large KL divergence $D_{KL}(Q \| P)$ in the PAC-Bayes bound, resulting in high test error. This is the 'overfitting' regime where both classical and modern theory agree that generalisation is poor.

Overparameterised regime ($p \gg n$): the minimum-norm interpolant has low norm $\|w^*\|$ (because there are many solutions to the interpolation problem, and the minimum-norm one has small norm distributed over many parameters). The KL divergence $D_{KL}(Q \| P)$ is small, giving a non-vacuous PAC-Bayes bound. Test error decreases as p increases, eventually matching the optimal minimax rate.

Theorem 7.5 (overparameterisation does not increase minimax risk) confirms that the minimax rate is the same for $p > n$ as for $p = n^*$ (the optimal model size). The double descent curve is therefore not a failure of the information-theoretic framework — it is a consequence of the shift from the empirical risk minimiser ($p < n$) to the minimum-norm interpolant ($p > n$), both achieving the same information-theoretic optimum.

7.5b Information-Theoretic Analysis of Large Language Models

Large language models (LLMs) such as GPT-4 and Claude are trained on trillions of tokens of text. The information-theoretic analysis of their generalisation requires extending the standard framework to the sequence prediction setting.

For autoregressive language models, the natural loss is the negative log-likelihood (cross-entropy) per token: $L(w) = (1/T)\, E[-\log p_w(x_t | x_{<t})]$. The IZS mutual information bound (Theorem 7.5) gives: $E[L(A(S)) - \hat{L}(A(S))] \leq \sqrt{(2I(A(S);S)/nT)}$, where nT is the total number of tokens in the training set.

For a language model with $p = 10^{11}$ parameters trained on $nT = 10^{13}$ tokens, the IZS bound requires $I(A(S);S) \leq O(nT) = O(10^{13})$ bits for the generalisation gap to be sub-constant. In practice, gradient descent may transmit much less information from the training set to the model: the effective $I(A(S);S)$ is determined by the bottleneck of the gradient compression achieved by SGD, which is approximately the number of gradient steps times the number of bits per step.

The information-theoretic lower bound on the generalisation of LLMs is the entropy rate of natural language: the minimum cross-entropy loss achievable on any natural language corpus. Current estimates of the entropy rate of English text are approximately 0.7–1.3 bits per character (Shannon 1951; Brown et al. 1992), or about 1–2 bits per token (depending on tokenisation). State-of-the-art LLMs achieve cross-entropy losses of approximately 1.5–2.5 bits per token on held-out text, suggesting they are within 0.5–1 bit per token of the fundamental limit.

8.3b The Compression-Prediction Duality in Practice

The compression-prediction duality (Theorem 8.2) has practical consequences for how we should train and evaluate deep learning systems.

Consequence 1 (The optimal model is the most compressed model that achieves the desired accuracy): given a target accuracy ε, the optimal model is the one that achieves accuracy ε with minimum model complexity (bits of description). This is the MDL perspective on model selection: prefer the smallest model that is consistent with the data.

Consequence 2 (Larger training sets allow more complex models): the minimax rate n^{-2s/(2s+d)} implies that as n increases, the optimal model complexity grows as n^{d/(2s+d)}. This explains why larger datasets benefit from larger models: more data can justify more complex representations.

Consequence 3 (Compression metrics predict generalisation): the KL divergence D_KL(Q ∥ P), the stochastic complexity SC(S|M), and the IZS mutual information I(A(S);S) all bound the generalisation gap. Measuring these quantities during training can predict generalisation without access to a held-out test set.

Consequence 4 (The optimal training algorithm is Gibbs sampling): the Gibbs algorithm A*(S) ∝ P e^{−βnL̂} is simultaneously optimal under all four frameworks. While exact Gibbs sampling is intractable, approximate Gibbs sampling via MCMC, Langevin dynamics, or stochastic gradient descent provides a tractable alternative. The effective temperature of SGD controls how close it is to the Gibbs algorithm.

These consequences provide actionable guidance for practitioners: use MDL-motivated regularisation, prefer flatter minima (SAM, SWA), and track the effective dimensionality of the model during training as a proxy for generalisation.

8.4b Information-Theoretic Connections to Scaling Laws

Scaling laws for language models (Kaplan et al. 2020; Hoffmann et al. 2022) describe the power-law relationship between model performance and model size, dataset size, and compute. The information-theoretic framework provides a theoretical foundation for these empirical laws.

The scaling law for test loss vs. model parameters p (with fixed training tokens nT): L(p) ≈ L* + A/p^α, where L* is the irreducible entropy (minimum achievable cross-entropy), A is a constant, and α is the scaling exponent. Empirically, α ≈ 0.076 for language models (Kaplan et al. 2020).

The information-theoretic explanation: as p increases, the model's effective dimensionality d_eff increases (more parameters can capture more structure in the data), reducing the approximation error. The scaling exponent α = 2s/(2s+d) from the minimax rate theorem gives α ≈ 0.076 if s/(s+d/2) ≈ 0.076 — consistent with d/(2s) ≈ 12, suggesting the effective data dimension is about 12 times the smoothness parameter.

The scaling law for test loss vs. training tokens nT (with fixed model size p): L(nT) ≈ L* + B/nT^β, where β ≈ 0.095 (Kaplan et al. 2020). The information-theoretic explanation:

as nT increases, the estimation error (which scales as $\sqrt{(I/nT)}$) decreases, reducing the generalisation gap. The exponent $\beta = 1/(2s+d)$ from the minimax rate theorem gives $\beta \approx 0.095$ if $2s+d \approx 10.5$, consistent with the estimate from α.

These connections suggest that scaling laws are not empirical coincidences but consequences of the information-theoretic structure of language modelling — specifically, the minimax rate theorem applied to the compositional structure of natural language (which can be decomposed into syntax, semantics, and pragmatics with different complexity levels).

9.3b DP-SGD in Practice: Hyperparameter Tuning

Implementing DP-SGD requires careful tuning of three hyperparameters: the clipping norm C, the noise multiplier σ, and the number of training epochs T. The choice of these hyperparameters determines the privacy-utility trade-off.

The clipping norm C should be set to approximately the median gradient norm: clip_norm ≈ median_i $\| \nabla_w \ell(f_w(x_i), y_i) \|_2$. Too small a clip norm clips most gradients, adding more noise than necessary; too large a clip norm doesn't clip enough, adding more noise than necessary to maintain privacy.

The noise multiplier σ should be chosen to achieve the desired ε after T training steps: $\sigma \approx \sqrt{(2T \log(1/\delta))} / (q\varepsilon\sqrt{n})$, where q = batch size / n is the sampling rate. Larger σ gives stronger privacy (smaller ε) but noisier gradient estimates (worse utility).

The number of epochs T is typically set to the minimum needed to achieve the desired utility. More epochs give better utility but consume more privacy budget (ε grows as $\sqrt{T}$). The optimal T balances utility improvement against privacy cost.

Practical results: on CIFAR-10 with $\varepsilon = 2$, $\delta = 10^{-5}$, DP-SGD achieves approximately 88% test accuracy (vs. 96% non-privately) with ResNet-9, using about 200 epochs of training. The 8% accuracy gap is the 'cost of privacy' for this setting. On MNIST, the gap is smaller: 98% DP vs. 99.5% non-DP at $\varepsilon = 1$.

9.4b Federated Learning and Differential Privacy

Combining federated learning with differential privacy (DP-FL) is the standard approach for training private machine learning models in practice. The key challenge is that each client's local update reveals information about its local dataset, which must be protected.

The standard DP-FL protocol (McMahan et al. 2018) applies DP-SGD locally on each client before transmitting the update to the server. The server aggregates the noisy updates and broadcasts the result to all clients. The privacy analysis must account for both the local noise added by DP-SGD and the privacy amplification from subsampling (each round only a fraction of clients participate).

Privacy amplification by subsampling: if each client participates in each round with probability q (and K clients participate out of a total of N, giving q = K/N), then the effective privacy guarantee is amplified: an algorithm that satisfies (ε, δ)-DP without

subsampling satisfies approximately (qε, qδ)-DP with subsampling. This amplification factor q can be substantial (e.g., q = 0.01 gives 100× amplification).

In practice, DP-FL with ε ≈ 8, δ = 10^{-6} and 100 clients achieves near-non-private accuracy for tasks with large amounts of local data (millions of local training examples per client, as in keyboard prediction at large tech companies). For tasks with limited local data (a few hundred examples per client, as in medical federated learning), the accuracy gap is larger and ε must be set larger to achieve acceptable utility.

10.3b The Natural Gradient and Second-Order Methods

Natural gradient descent is the continuous-time limit of Newton's method applied to the loss function on the statistical manifold. To see this, consider the second-order Taylor expansion of D_KL(p_w+Δw ∥ p_w) in the weight perturbation Δw:

$$D_{KL}(p_{w+\Delta w} \,\|\, p_w) \approx \tfrac{1}{2} \Delta w^{T} I_F(w) \Delta w.$$

The natural gradient direction is the direction that maximally increases the loss L(w) subject to a constraint on D_KL(p_{w+Δw} ∥ p_w) ≤ ε. By Lagrange multipliers, this gives $\Delta w^* \propto -I_F(w)^{-1} \nabla_w L(w) = -\tilde{\nabla}_w L(w)$. Natural gradient descent is therefore the Riemannian gradient descent on the statistical manifold with the Fisher metric, and it corresponds to steepest descent in the metric of the output distribution space rather than the parameter space.

The connection to second-order methods: Newton's method uses the Hessian $\nabla^2_w L(w)$ to precondition the gradient. At a minimum w*, $\nabla^2_w L(w^*) = I_F(w^*)$ (Proposition 10.2), so Newton's method and natural gradient descent agree at the optimum. Away from the optimum, they differ because $\nabla^2_w L(w) \neq I_F(w)$ in general. Natural gradient descent is 'safer' than Newton's method: it uses the Fisher matrix (always positive semidefinite) rather than the Hessian (which may be indefinite away from a minimum), avoiding divergence due to negative curvature directions.

Approximate natural gradient methods: K-FAC (Martens and Grosse 2015) approximates I_F(w) by a block-diagonal matrix where each block corresponds to one layer. Within each block, I_F is approximated by a Kronecker product of the input covariance and the gradient covariance: $I_F^{(\ell)} \approx A_\ell \otimes G_\ell$, where $A_\ell = \mathbb{E}[a_{\ell-1} a_{\ell-1}^T]$ (input activation covariance) and $G_\ell = \mathbb{E}[g_\ell g_\ell^T]$ (pre-activation gradient covariance). The inverse of this Kronecker product is $(A_\ell \otimes G_\ell)^{-1} = A_\ell^{-1} \otimes G_\ell^{-1}$, which can be computed efficiently.

Empirical performance: K-FAC achieves 2–5× fewer gradient steps than Adam to reach the same training loss for large language models, at the cost of approximately 3× more memory. For very large models ($p > 10^9$), the memory overhead of K-FAC becomes prohibitive, and diagonal approximations (Adam) or parameter-efficient variants (LoRA) are preferred.

10.4b The Neural Tangent Kernel and Feature Learning

The Neural Tangent Kernel (NTK) regime describes the dynamics of infinitely wide networks trained by gradient descent. In this regime, the NTK matrix $K(x, x') = E_{w\sim P_0}[\nabla_w f_w(x) \cdot \nabla_w f_w(x')]$ remains constant throughout training, and the network function evolves linearly:

$$f_{w_t}(x) = f_{w_0}(x) - K(x, X_{train})(K(X_{train}, X_{train}) + \lambda I)^{-1} (f_{w_0}(X_{train}) - Y) \cdot (1 - e^{-\lambda t}).$$

This is exactly kernel ridge regression with the NTK as the kernel, confirming that infinitely wide networks are equivalent to kernel methods.

The NTK regime does not feature learn: the representations learned by each layer remain frozen at their initialisation, and only the top layer is effectively trained (through the linear predictor). This is in contrast to finite-width networks, which feature learn: the representations change during training, adapting to the specific task.

Information-geometric interpretation: in the NTK regime, the statistical manifold is linear (the network function space is a flat subspace of all measurable functions under the NTK metric), and natural gradient descent reduces to gradient descent in the kernel function space. Feature learning corresponds to moving away from the initialisation point in the statistical manifold — a non-linear movement that the NTK approximation cannot capture.

The transition from NTK to feature learning regime: as the width decreases or the learning rate increases, the network exits the NTK regime and begins to feature learn. The critical learning rate $\eta^* \approx 2/(\lambda_{max} + \lambda_{min})$ (where λ_{max}, λ_{min} are the largest and smallest eigenvalues of the NTK matrix) marks the transition. For $\eta < \eta^*$, the network stays in the NTK regime; for $\eta > \eta^*$, feature learning occurs.

11.3b Non-IID Data and the Information-Theoretic Analysis of Heterogeneity

The most challenging aspect of federated learning in practice is data heterogeneity: the local distributions D_k differ across clients. In the extreme case, each client has data from a single class (perfect label skew), making local gradients maximally heterogeneous and local model updates nearly orthogonal to the global gradient.

The information-theoretic analysis of heterogeneity: define the heterogeneity of the federated learning problem as $\varkappa = (1/K)\ \Sigma_k\ D_{KL}(D_k \ /\!/\ D)$, where $D = (1/K)\Sigma_k\ D_k$ is the global distribution. For $\varkappa = 0$ (IID data), local updates are unbiased estimates of the global gradient. For $\varkappa > 0$, local updates are biased, and the bias grows with $\varkappa$.

The federated PAC-Bayes bound (Theorem 11.2) degrades as the aggregated KL $(1/K)\Sigma_k\ D_{KL}(Q_k /\!/ P)$ increases. This KL is controlled by both the learning rate and the number of local steps: more local steps → more deviation from the prior P → higher KL → worse generalisation bound.

The optimal number of local steps per round: let E be the number of local SGD steps per client per round. The communication efficiency improves as E increases (fewer rounds needed), but the gradient bias also increases with E (more deviation from the

global optimum). The optimal E balances communication efficiency against gradient bias:

$$E^* = \text{argmin}_E \left[T(E) \cdot B + n/E \cdot \eta \cdot \varkappa \cdot \| \nabla L(w^*) \|^2 \right],$$

where T(E) is the number of rounds needed for convergence (decreasing in E) and the second term is the gradient bias cost (increasing in E). For strongly heterogeneous federated problems (large $\varkappa$), $E^* \approx 1$ (one local step per round, equivalent to minibatch SGD with communication after each step). For homogeneous problems (small $\varkappa$), E^* can be much larger.

11.4b Communication Efficiency and Information Theory

The fundamental question of federated learning is: how much communication is needed to achieve a given accuracy? The answer depends on the information-theoretic structure of the gradient signal and the communication budget.

The gradient signal has a natural information-theoretic structure: the gradient $\nabla_w L(w_t)$ is a p-dimensional vector that can be compressed without significant loss. The key observation is that gradients are approximately low-rank: for a network with L layers and weight matrices $W_1, \ldots, W_L$, the gradient $\nabla_{W_\ell} L \approx a_\ell g_\ell^T$ (rank-1 for a single example, low-rank for a minibatch). Low-rank gradients can be transmitted with far fewer bits than full-rank vectors.

Rank-r gradient compression: instead of transmitting the full gradient $\nabla_w L$ (p bits at 32 bits per parameter), transmit its rank-r approximation $\text{SVD}_r(\nabla_w L)$ ($2rp/\max(p_1,p_2)$ parameters for a $p_1 \times p_2$ weight matrix, roughly rp bits). For $r = O(1)$ (batch size = 1), this reduces communication by a factor of $O(\min(p_1,p_2)/r)$ — potentially orders of magnitude.

PowerSGD (Vogels et al. 2019): a practical rank-r gradient compression algorithm that compresses the gradient of each weight matrix to its best rank-r approximation. With $r = 2$ (rank-2 compression), PowerSGD achieves within 1% of uncompressed SGD accuracy on image classification tasks while reducing communication by 100–1000×.

The information-theoretic limit: the minimum bits to transmit the rank-r gradient approximation for a random $p_1 \times p_2$ matrix drawn from a Gaussian distribution with covariance Σ is given by the water-filling rate-distortion function: $B_{\min}(D) = \sum_{i=1}^{r} (1/2) \log(\sigma_i^2/\lambda)$, where σ_i are the singular values of the gradient and λ is chosen to achieve distortion D. PowerSGD approaches this limit within a constant factor for practical gradient distributions.

1.A Information Theory Foundations

Before developing the IB in depth, we review the foundational information-theoretic quantities that appear throughout this volume. All quantities are defined for jointly distributed random variables (X, Y) with joint distribution p(x, y) over alphabets X and Y.

The Shannon entropy $H(X) = -\Sigma_x p(x) \log p(x)$ measures the average uncertainty in X. It equals the minimum expected code length for X under any prefix-free code, by Shannon's source coding theorem. Entropy is always non-negative, is maximised by the uniform distribution ($H_{max} = \log |X|$), and equals 0 if and only if X is deterministic.

The conditional entropy $H(X|Y) = \Sigma_y p(y) H(X|Y=y) = -E_{p(x,y)}[\log p(x|y)]$ measures the average uncertainty in X given Y. By the chain rule: $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$. Conditioning can only reduce entropy: $H(X|Y) \leq H(X)$, with equality if and only if $X \perp Y$.

The mutual information $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y)$ measures the statistical dependence between X and Y. Key properties:

- Non-negativity: $I(X;Y) \geq 0$, with equality if and only if $X \perp Y$.
- Symmetry: $I(X;Y) = I(Y;X)$.
- Data processing inequality: for any Markov chain $X \to T \to Y$: $I(X;Y) \geq I(T;Y)$ and $I(X;Y) \geq I(X;T)$. Applying functions to X or Y can only decrease MI.
- Chain rule: $I(X;Y,Z) = I(X;Y) + I(X;Z|Y)$. MI decomposes into a sum of conditional MIs.
- For jointly Gaussian (X,Y): $I(X;Y) = -(1/2) \log(1 - \rho^2)$, where ρ is the correlation coefficient.

The Kullback-Leibler divergence $D_{KL}(P \,/\!/\, Q) = \Sigma_x P(x) \log(P(x)/Q(x))$ measures the 'information loss' when using distribution Q to model distribution P. $D_{KL}(P \,/\!/\, Q) \geq 0$ (Gibbs' inequality), with equality if and only if $P = Q$. D_{KL} is not symmetric: $D_{KL}(P \,/\!/\, Q) \neq D_{KL}(Q \,/\!/\, P)$ in general. The mutual information can be written as $I(X;Y) = D_{KL}(p(x,y) \,/\!/\, p(x)p(y))$ — the KL divergence between the joint and the product of marginals.

For continuous random variables, the differential entropy $h(X) = -\int p(x) \log p(x)\, dx$ replaces the discrete entropy. Unlike discrete entropy, differential entropy can be negative (e.g., $h(N(0,\sigma^2)) = (1/2) \log(2\pi e\sigma^2)$, which is negative for $\sigma < 1/\sqrt{(2\pi e)}$). The mutual information $I(X;Y) = h(X) + h(Y) - h(X,Y)$ remains non-negative even for continuous variables, because the differential entropy terms cancel.

1.B The Data Processing Inequality: Proofs and Extensions

The data processing inequality (DPI) is the most fundamental result in information theory for understanding representation learning. We present a complete proof and several extensions.

Theorem 1.A (Data Processing Inequality).
For any Markov chain $X \to T \to Y$ (i.e., X and Y are conditionally independent given T): $I(X;Y) \geq I(T;Y)$ and $I(X;Y) \geq I(X;T)$.
Moreover, $I(X;T) = I(X;Y)$ if and only if T is a sufficient statistic of X for Y.

Proof of $I(X;Y) \geq I(T;Y)$. By the chain rule for MI:

$$I(X;Y,T) = I(X;T) + I(X;Y|T) = I(X;Y) + I(X;T|Y).$$

Since $X \to T \to Y$ is a Markov chain, $I(X;Y|T) = 0$ (knowing T makes X and Y conditionally independent). Therefore $I(X;T) + 0 = I(X;Y) + I(X;T|Y)$, giving $I(X;Y) = I(X;T) - I(X;T|Y)$. Since $I(X;T|Y) \geq 0$, we have $I(X;Y) \leq I(X;T)$. Wait — this proves $I(X;T) \geq I(X;Y)$, not $I(X;Y) \geq I(T;Y)$. Let us redo.

We use $I(X;Y,T) = I(X;Y) + I(X;T|Y) = I(X;T) + I(X;Y|T)$. Since $X \to T \to Y$: $I(X;Y|T) = 0$, so $I(X;T) = I(X;Y) + I(X;T|Y) \geq I(X;Y)$. Also, $I(T;Y,X) = I(T;X) + I(T;Y|X) = I(T;Y) + I(T;X|Y)$. Since $T = f(X)$, $I(T;Y|X) = 0$, so $I(T;X) = I(T;Y) + I(T;X|Y) \geq I(T;Y)$. And $I(T;X) = I(X;T) \geq I(X;Y)$ (from above), so $I(X;Y) \geq I(X;Y)$ — hmm, we need to connect these. Since $I(X;T) \geq I(X;Y)$ and $I(X;T) = I(T;X) \geq I(T;Y)$, combining: $I(X;Y) \leq I(X;T)$ and $I(T;Y) \leq I(X;T)$. But does $I(X;Y) \geq I(T;Y)$? Yes: $I(T;Y) \leq I(X;T)$ (DPI in the chain $X \to T$ with Y as 'parameter') and the chain $Y \to T \to X$ gives $I(Y;X) \geq I(T;X) \geq I(T;...$ We can directly use: in the chain $Y \leftrightarrow X \to T$, the DPI gives $I(X;Y) \geq I(T;Y)$. □

Extensions of the DPI:

- Stochastic maps: for any stochastic map $p(t|x)$ (not necessarily deterministic), $I(X;T) \geq I(f(X);T)$ for any function f of X. Determinism is not needed for the DPI.
- Multiple observations: for i.i.d. observations $T_1, \ldots, T_n$ of X, $I(X; T_1, \ldots, T_n) \leq n \cdot I(X; T_1)$ (sub-additivity of MI). Equality holds if and only if the T_i are conditionally independent given X.
- Vector observations: for $T = (T_1, T_2)$, $I(X;T) = I(X;T_1) + I(X;T_2|T_1) \geq I(X;T_1)$. Adding observations can only increase MI.

1.C The IB from the Rate-Distortion Perspective

The connection between the IB and rate-distortion theory (developed fully in Chapter 6) is best understood through the channel coding perspective. A stochastic encoder $p(t|x)$ defines a channel from X to T with capacity $C = \max_{p(x)} I(X;T)$. The IB asks: what is the minimum channel rate $R = I(X;T)$ needed to transmit I_0 bits of information about Y through the channel?

This is precisely a rate-distortion problem: the 'source' is X, the 'reconstruction' is T, and the 'distortion' measure is $d(x,t) = D_{KL}(p(y|x) \,/\!/\, p(y|t))$ — the cost of approximating the conditional distribution of Y given x by the conditional distribution of Y given t. The rate-distortion function for this distortion measure is:

$$R(D) = \min_{p(t|x):\, \mathbb{E}[D_{KL}(p(y|x) \,/\!/\, p(y|t))] \leq D} I(X;T).$$

The IB curve $I^*(R) = \max_{p(t|x):\, I(X;T) \leq R} I(T;Y)$ is the complement of R(D): instead of asking the minimum rate to achieve distortion D, it asks the maximum relevance $I(T;Y)$ achievable at compression rate R. The two curves are related by $I^*(R) = H(Y) - D^*(R)$, where $D^*(R) = \min_{p(t|x):\, I(X;T) \leq R} H(Y|T)$ is the minimum conditional entropy of Y given the representation.

This connection has a practical implication: the IB can be solved by any rate-distortion algorithm (e.g., Blahut-Arimoto, Chapter 6) with the KL divergence as the distortion measure. The IB-specific fixed-point equations (Theorem 1.1) are the stationarity conditions of the Blahut-Arimoto algorithm for this specific distortion measure.

1.D The Information Bottleneck and Sufficient Statistics: A Deeper Analysis

The connection between the IB and sufficient statistics is deeper than it might first appear. We develop this connection in detail, showing how the IB generalises sufficiency to the case where the 'target' is a related random variable Y rather than a parameter θ.

Classical sufficiency: T is sufficient for θ if $H(X|T, \theta) = H(X|T)$ — knowing θ adds no information about X beyond what T already contains. This is equivalent to the Markov property $\theta \leftrightarrow T \leftrightarrow X$: the parameter θ is conditionally independent of the raw data X given the sufficient statistic T.

IB sufficiency: T is an IB-sufficient statistic for Y at level β if $p^*(t|x) = p^*(t|x; \beta)$ — the IB-optimal encoder at trade-off parameter β. The IB-sufficient statistic is not the same as the classical sufficient statistic unless $\beta \to \infty$. For finite β, the IB-sufficient statistic is a 'soft' sufficient statistic that maximally compresses X while retaining β bits of information about Y.

The IB phase transitions (Theorem 1.2) correspond to the emergence of new 'relevant directions' in the data as β increases. At each critical value β_k, a new linear combination of the data becomes relevant for predicting Y — exactly the sequence of sufficient statistics that would emerge from applying CCA at progressively weaker regularisation.

Proposition 1.A (IB and Linear Sufficient Statistics).
For jointly Gaussian (X,Y) and linear encoders $T = AX + \text{noise}$, the IB-optimal encoder at trade-off β uses the top-k singular directions of $\Sigma_{XX}^{-1/2} \Sigma_{XY}$, where k is the number of singular values σ_i satisfying $\beta \cdot \sigma_i > 1$.
The critical values $\beta_k = 1/\sigma_k$ (reciprocals of singular values) correspond to the emergence of the k-th canonical direction, matching the classical canonical correlation analysis solution.

This proposition establishes a precise correspondence between the IB phase transitions and the canonical correlations of the joint Gaussian distribution. The k-th critical value β_k is the reciprocal of the k-th canonical correlation $\sigma_k = \rho_k$ (the k-th largest canonical correlation coefficient). At $\beta = \beta_k$, the k-th canonical direction becomes 'worth encoding' — the marginal benefit of adding this direction (in bits of I(T;Y) per bit of I(X;T)) equals β_k.

2.A PAC-Bayes Theory: Historical Context and Motivation

The PAC-Bayes framework was developed independently by McAllester (1999) and Seeger (2002), building on earlier work by Shawe-Taylor and Williamson (1997). The key innovation was the shift from worst-case bounds over a fixed hypothesis class to

average-case bounds over a distribution on hypotheses, incorporating prior knowledge through the prior distribution P.

The motivation for PAC-Bayes comes from the observation that classical PAC learning bounds are vacuous for modern neural networks. VC dimension bounds scale as $O(\sqrt{(d/n)})$ where d is the VC dimension — for a p-parameter network, $d \geq p$, giving bounds of $O(\sqrt{(p/n)}) \approx O(1)$ for $p = n$ (overparameterised regime). Rademacher complexity bounds face the same problem.

The key insight of PAC-Bayes is that the VC dimension bounds all hypotheses simultaneously, including adversarially chosen ones. A learning algorithm like SGD does not return an adversarial hypothesis — it returns a specific hypothesis shaped by the training data and the algorithm's implicit biases. PAC-Bayes exploits this structure by bounding the complexity of the returned hypothesis relative to the prior, not the complexity of the entire hypothesis class.

The change-of-measure inequality at the heart of PAC-Bayes ($E_{h\sim Q}[\phi(h)] \leq D_{KL}(Q \,/\!/\, P) + \log E_{h\sim P}[e^{\phi(h)}]$) is a consequence of the Gibbs variational principle in statistical mechanics: the free energy $F = \inf_Q [E_Q[E] + T^{-1} D_{KL}(Q \,/\!/\, P)]$ is achieved by the Gibbs distribution $Q^*(h) \propto P(h)\, e^{-E(h)/T}$. Setting $\phi = -E/T$ and using the variational principle gives the PAC-Bayes change-of-measure inequality directly.

2.B The Catoni Bound: Derivation and Optimisation

The Catoni bound (Theorem 2.3) is derived using a different concentration inequality from the McAllester-Seeger bound. The key lemma is:

Lemma 2.A (Catoni's Concentration Lemma).
For any random variable Z with $E[Z] = \mu$, if $Z \in [0,1]$, then for any $\lambda > 0$ and any $\delta \in (0,1)$, with probability $\geq 1 - \delta$:
$\mu \leq (1/(1 - e^{-\lambda/n})) \cdot [(1/n)\, \Sigma_i\, Z_i + (\log(1/\delta))/\lambda]$.
The function $(1/(1-e^{-\lambda/n}))$ is a Catoni-type correction factor that improves on the Hoeffding bound by a factor of approximately 2 for small λ/n.

Applying this lemma to $Z = \ell(h, X_i) \in [0,1]$ and integrating over $h \sim Q$ gives the Catoni bound. The optimal λ^* minimises the right-hand side, and the resulting bound is:

$$L(Q) \leq \hat{L}(Q) + \sqrt{(\hat{L}(Q) \cdot D_{KL}(Q \,/\!/\, P)/n)} + D_{KL}(Q \,/\!/\, P)/n + O(\log(1/\delta)/n).$$

The key improvement over McAllester-Seeger is the $\sqrt{(\hat{L}(Q) \cdot D_{KL}(Q \,/\!/\, P)/n)}$ term, which is smaller than the McAllester-Seeger bound of $\sqrt{(D_{KL}(Q \,/\!/\, P)/n)}$ whenever $\hat{L}(Q) < 1$. For modern deep networks trained to near-zero training loss ($\hat{L}(Q) \approx 0$), the Catoni bound gives:

$$L(Q) \leq O(\sqrt{(\hat{L}(Q) \cdot D_{KL}(Q \,/\!/\, P)/n)}) \to 0 \quad \text{as} \quad \hat{L}(Q) \to 0,$$

which is much tighter than the McAllester-Seeger bound of $O(\sqrt{(D_{KL}(Q \,/\!/\, P)/n)})$ that does not vanish with $\hat{L}(Q)$.

The Catoni bound is the tightest available PAC-Bayes bound for the bounded-loss setting. For unbounded losses (e.g., squared error regression), the Catoni bound requires additional assumptions (sub-Gaussianity or moment bounds) and the bound takes a different form.

2.C PAC-Bayes for Stochastic Classifiers

The Gibbs classifier (Definition 2.1) draws h ~ Q and returns h(x). This is a stochastic classifier: on the same input x, it may return different predictions on different runs. In practice, deterministic classifiers are preferred (they give reproducible predictions), but the PAC-Bayes framework naturally applies to stochastic classifiers.

The expected 0-1 loss of the Gibbs classifier under posterior Q is: L_{0-1}(Q) = E_{h~Q}[P(h(X) ≠ Y)] = P(h(X) ≠ Y) where h is drawn fresh from Q for each test example. This is a 'majority vote' classifier in expectation: it is equivalent to the deterministic classifier that predicts argmax_y P_{h~Q}(h(X) = y).

The PAC-Bayes bound for the 0-1 loss has a different form from the cross-entropy bound because the 0-1 loss is not differentiable. The relevant result is:

Theorem 2.A (PAC-Bayes Bound for 0-1 Loss).
For any prior P and posterior Q, with probability ≥ 1 − δ over S:
kl(L̂_{0-1}(Q) ‖ L_{0-1}(Q)) ≤ (D_KL(Q ‖ P) + log(2√n/δ)) / n,
where kl(q ‖ p) = q log(q/p) + (1-q) log((1-q)/(1-p)) is the binary KL divergence.
For L̂_{0-1}(Q) < 1/2 (better than random), this implies: L_{0-1}(Q) ≤ h^{-1}_+(L̂_{0-1}(Q) + √(D_KL(Q ‖ P)/n)),

where h^{-1}_+(q) ≤ q + √(q(1-q) · D_KL(Q ‖ P)/n) + ... The binary KL bound is tighter than the Hoeffding-based bound for small error rates, and it is the bound used by Dziugaite and Roy (2017) to obtain the first non-vacuous bounds for neural networks.

2.D Symmetrised PAC-Bayes and the Ghost Sample

The PAC-Bayes framework requires choosing the prior P before observing the training data S. This can be overly restrictive: in practice, we often have knowledge about the problem (e.g., the network architecture, the typical scale of weights) that we want to encode in P, but this knowledge may be data-dependent.

The ghost sample technique (Theorem 2.6) allows a data-dependent prior by splitting the training data into two halves. A more elegant alternative is the symmetrised PAC-Bayes bound, which uses all n training examples for both the prior computation and the bound, at the cost of a slightly worse O(√(1/n)) rate.

Theorem 2.B (Symmetrised PAC-Bayes Bound).
Let $\mathcal{A}$ be any learning algorithm and P_S the prior computed by running $\mathcal{A}$ on a 'ghost' training set S' drawn independently from D^n. Let Q be the posterior computed by running $\mathcal{A}$

on the actual training set S. Then with probability $\geq 1 - \delta$ over the joint draw of S, $S' \sim D^n$:
$L(Q) \leq \hat{L}(Q) + \sqrt{[(D_{KL}(Q \,\|\, P_S) + \log(2/\delta)) / n]}$.
The bound holds despite P_S being data-dependent, because S and S' are independent.

The symmetrised bound is useful when the prior P_S is a meaningful approximation to the posterior Q: if A initialises near the optimal solution, P_S and Q_S will be close in KL divergence, giving a tight bound. In the extreme case where A is deterministic and returns the same solution for any input dataset (no dependence on the training data), $P_S = Q$ and $D_{KL}(Q \,\|\, P_S) = 0$, giving a trivial bound — which is correct, because a deterministic algorithm that doesn't use the data doesn't generalise.

3.A Kraft's Inequality and the Relationship Between Codes and Distributions

The foundation of MDL is the bijection between prefix-free codes and probability distributions, established by Kraft's inequality and Shannon's source coding theorem. This bijection makes precise the sense in which 'shorter code = higher probability'.

Theorem 3.A (Kraft's Inequality).
A set of code lengths $\{L(x) : x \in X\}$ corresponds to a valid prefix-free code if and only if $\Sigma_x 2^{-L(x)} \leq 1$. The lengths $L(x) = -\log_2 P(x)$ for any probability distribution P over X satisfy Kraft's inequality, giving the Shannon code with expected length $E[L(X)] = H(X)$ (in bits).
Conversely, any valid code lengths $\{L(x)\}$ define a probability distribution $P_L(x) = 2^{-L(x)} / \Sigma_{x'} 2^{-L(x')}$, the normalised version of the Boltzmann weights. The MDL prior corresponding to code L is P_L.

This bijection between codes and distributions is the key to understanding MDL: any prior distribution P over hypotheses corresponds to a code with lengths $-\log P(h)$, and any code corresponds to a prior. The choice of prior is equivalent to the choice of code, and vice versa.

The universality of the NML code (Theorem 3.1) can now be interpreted: the NML code is the unique code that minimises the maximum regret (the maximum difference between the actual code length and the code length that would have been used with knowledge of the data-generating parameter). The NML code hedges against adversarial data by assigning shorter code lengths to sequences that would be more compressible under any model in M.

Practical construction of NML codes: for exponential families, the NML distribution $p_{NML}(x^n) = p(x^n \mid \hat{\theta}(x^n)) / C_n(M)$ is a proper probability distribution over X^n. The NML code assigns length $-\log_2 p_{NML}(x^n) = -\log_2 p(x^n \mid \hat{\theta}(x^n)) + \log_2 C_n(M)$ bits to the sequence x^n. This code is optimal in the minimax regret sense, but computing $C_n(M)$ may be intractable. The BIC approximation $\log_2 C_n(M) \approx (d/2) \log_2 n$ provides a tractable approximation.

3.B MDL and Model Averaging

Two-part MDL selects a single best model (the MDL estimator), but Bayesian model averaging considers all models simultaneously, weighting by their posterior probability. The MDL perspective on model averaging uses the NML distribution as the weighting distribution.

Bayesian model averaging: given a set of model classes $M_1, M_2, \ldots, M_K$ with prior probabilities $P(M_k)$ and evidence $p(S \mid M_k) = \int p(S \mid \theta, M_k)\, p(\theta \mid M_k)\, d\theta$, the Bayesian model average prediction is:

$$p_{BMA}(y \mid x, S) = \Sigma_k P(M_k \mid S)\, p(y \mid x, \hat{\theta}_{M_k}),$$

$$P(M_k \mid S) \propto P(M_k) \cdot p(S \mid M_k).$$

The MDL perspective replaces the evidence $p(S \mid M_k)$ with the NML likelihood $p_{NML}(S \mid M_k) = p(S \mid \hat{\theta}(S), M_k) / C_n(M_k)$. The MDL model weights are:

$$P_{MDL}(M_k \mid S) \propto P(M_k) \cdot p_{NML}(S \mid M_k) \propto P(M_k) \cdot p(S \mid \hat{\theta}(S), M_k) / C_n(M_k).$$

This gives a principled, non-Bayesian model averaging scheme that penalises model complexity through the parametric complexity $C_n(M_k)$. Models with high complexity (many parameters, high Fisher information volume) receive lower weight, even if they fit the data equally well.

In deep learning, MDL model averaging corresponds to ensembling networks of different sizes: a large ensemble of small networks is penalised by each network's complexity, while a single large network is penalised by its higher complexity. The optimal ensemble size balances fitting error against complexity penalty, giving a principled basis for ensemble design.

3.C MDL for Sequential Data: Language Models

Language modelling — predicting the next word in a sequence given the preceding context — is naturally formulated as a prequential MDL problem. The language model is trained to minimise the prequential code length:

$$L_{prequential}(w_1, w_2, \ldots, w_T) = \Sigma_{t=1}^{T} [-\log_2 p_\theta(w_t \mid w_{<t})],$$

where w_t is the t-th word (or token) and $w_{<t} = (w_1, \ldots, w_{t-1})$ is the preceding context. The prequential code length equals the total cross-entropy loss of the language model on the corpus, measured in bits.

The Shannon entropy rate of natural language $h_\infty = \lim_{n\to\infty} (1/n)\, H(W_1, \ldots, W_n)$ is the theoretical minimum prequential code length per token. Shannon (1951) estimated $h_\infty \approx 0.6$–1.3 bits per character for English text using a human prediction experiment. Modern estimates based on large language model perplexity give $h_\infty \approx 1.0$–1.5 bits per token (for byte-pair encoding with vocabulary size 50,257).

The bits-per-byte (BPB) or bits-per-character (BPC) metric used to evaluate language models is exactly the prequential code length per character. State-of-the-art language

models achieve BPC ≈ 0.9–1.1 on standard benchmarks (enwiki8, text8), approaching the estimated entropy rate of English text. The remaining gap between the achieved BPC and h_∞ represents the model's failure to capture long-range dependencies and semantic structure in the text.

MDL model selection for language models: given two language models M_1 (smaller, simpler) and M_2 (larger, more complex), the MDL criterion selects M_1 if its prequential code length $L_{M_1} < L_{M_2}$: $SC(w_1^T | M_1) = L_{M_1} + \log_2 C_T(M_1) < L_{M_2} + \log_2 C_T(M_2) = SC(w_1^T | M_2)$. For modern language models with $p \gg T$ (overparameterised), the complexity term $\log_2 C_T(M) \approx (p/2) \log T$ is proportional to the model size, giving a strong penalty for large models. In practice, the perplexity improvement from larger models often outweighs the MDL complexity penalty, which is why scaling laws favour larger models for larger datasets.

3.D The Minimum Message Length Principle

Minimum Message Length (MML, Wallace and Boulton 1968; Wallace and Freeman 1987) is an alternative information-theoretic approach to model selection and parameter estimation that is closely related to MDL but differs in its treatment of parameter precision.

Two-part MML: encode the parameter θ with precision ε (using $\log_2(1/\varepsilon)$ bits for each continuous parameter) and encode the data x^n given the quantised parameter θ_ε. The total message length is:

$$MML(x^n, \theta; \varepsilon) = -\log_2 P(\theta_\varepsilon) + (-\log_2 p(x^n | \theta_\varepsilon)),$$

where the optimal ε is chosen to minimise the total message length. The MML estimate minimises $MML(x^n, \theta; \varepsilon)$ over both θ and ε.

The key difference between MDL and MML: MDL (in the NML formulation) treats the entire model class M as the unit of description, while MML treats the individual parameter θ as the unit of description. MDL asks 'which model class best describes the data?' while MML asks 'which parameter value, at what precision, best describes the data?'

Relationship to MDL: the MML estimate converges to the MDL estimate (NML) in the limit of large n. For finite n, MML provides a correction that depends on the prior $P(\theta)$ and the Fisher information $I(\theta)$, giving a penalised likelihood estimator:

$$\hat{\theta}_{MML} = \arg\max_\theta \left[\log P(\theta) + \log p(x^n | \theta) - (d/2) \log(I(\theta)/(2\pi))\right].$$

This is the MDL estimator with the Jeffreys prior included explicitly in the likelihood term — a fully Bayesian interpretation of MDL that avoids the improper Jeffreys prior problem.

4.A The MINE Estimator: Variance Reduction and Bias Correction

The MINE estimator (Definition 4.1) has bias O(1/n) introduced by Jensen's inequality. For small batch sizes (n < 1000), this bias can be significant and must be corrected. Several variants of MINE address this.

MINE-f (f-MINE): replaces the log-mean with a directly estimated cumulant. For the function f* = log(dP/dQ) + const that achieves the DV supremum, the correction is:

$$\hat{I}_MINE\text{-}f(\phi) := E_P[f_\phi] - (E_Q[e^{f_\phi}] - 1) / E_Q[e^{f_\phi}] \cdot E_Q[e^{f_\phi}].$$

This correction uses a moving average of E_Q[e^{f_ϕ}] across batches, reducing the variance of the denominator estimate and thereby reducing the Jensen's inequality bias. MINE-f achieves lower variance than MINE at the same batch size, at the cost of tracking an additional running average.

Smoothed MINE: replaces the batch-level log-mean with a smoothed estimate that uses an exponential moving average (EMA) of E_Q[e^f]:

$$v_t = (1\text{-}a)\, v_\{t\text{-}1\} + a \cdot (1/n_t) \Sigma_j e^{f_\phi(x_j, \tilde{t}_j)},$$

$$\hat{I}_smooth(X;T;\phi) := (1/n) \Sigma_i f_\phi(x_i,t_i) - log(v_t).$$

The EMA parameter $\alpha \in (0,1)$ controls the trade-off between bias (smaller $\alpha \to$ lower variance but higher bias, as old batches contribute more) and variance (larger $\alpha \to$ lower bias but higher variance). In practice, $\alpha = 0.01$ provides a good balance.

InfoNCE (Contrastive Predictive Coding, van den Oord et al. 2018): uses K negative samples instead of one, giving a lower variance but biased-downward estimator:

$$\hat{I}_NCE(X;T;\phi) := (1/n) \Sigma_i [f_\phi(x_i,t_i) - log((1/K) \Sigma_j e^{f_\phi(x_i,\tilde{t}\square)})] \le log\ K.$$

InfoNCE is bounded by log K (the number of negative samples), so it can only estimate MI up to log K bits. For contrastive learning applications where high-MI representations are desired, using K = 65536 (as in SimCLR) allows estimation of MI up to log 65536 ≈ 16 bits.

4.B The Information Plane: Experimental Evidence and Reanalysis

The information plane was introduced by Tishby and Schwartz-Ziv (2017) as a diagnostic tool for visualising the learning dynamics of deep networks. We review the experimental evidence and the reanalysis that overturned the original claims.

Original experiment (Tishby-Schwartz-Ziv 2017): a fully-connected network with 7 hidden layers (12, 10, 7, 5, 4, 3, 2 neurons) was trained on a synthetic dataset with 4096 inputs and a binary target. Mutual information was estimated by binning activations into 30 equally spaced bins. Two-phase dynamics were observed: I(T;Y) increased rapidly in the first ~400 epochs (fitting phase), then I(X;T) decreased gradually (compression phase).

Saxe et al. (2018) reanalysis: the same network was trained with ReLU activations instead of tanh. No compression phase was observed with any estimator (binned or continuous). The compression phase with tanh activations was reproduced only with the binning

estimator; using KSG or MINE estimators, no compression was observed even for tanh networks.

Goldfeld and Polyanskiy (2020) theoretical analysis: proved that $I(X;T) = +\infty$ for deterministic networks with continuous inputs, making the binning estimator's 'finite' estimate a mathematical artefact. The noise-regularised MI $I_\sigma(X;T)$ was proposed as a theoretically grounded alternative.

Amjad and Geiger (2018) empirical study: systematically varied the activation function (ReLU, tanh, sigmoid, hard tanh), the binning resolution (5 to 50 bins), and the network depth (2 to 7 layers) on several datasets. Found:

- Compression phase appears only for saturating activations (tanh, sigmoid) with coarse binning.
- No compression with ReLU regardless of binning resolution.
- Compression timing correlates with saturation onset, not with the learning phase.
- The information plane position of layers is monotone with depth (deeper layers have lower MI on both axes) but the dynamics are not two-phase for non-saturating activations.

Current consensus: the two-phase dynamics are an artefact of the binning estimator and saturating activations. The information plane is a useful diagnostic tool when used with proper estimators (KSG, MINE, noise injection), but it does not show the two-phase dynamics claimed by Tishby-Schwartz-Ziv. The information plane under noise injection (Definition 4.3) is a theoretically well-founded diagnostic that shows meaningful dynamics without the artefacts.

4.C Practical Guide to MI Estimation for Deep Network Diagnostics

Based on the theoretical developments in this chapter and the experimental evidence reviewed in Section 4.B, we provide a practical guide to MI estimation for deep network diagnostics.

Step 1: Choose the representation dimensionality.

If the layer dimensionality $d^\ell > 50$, use PCA to reduce to $d_{PCA} = 10$–20 dimensions before estimating MI. The PCA projection retains most of the variance while making MI estimation tractable. Verify that the PCA projection preserves the MI by checking that $I_{KSG}(T^\ell_{PCA}; Y) \approx I_{KSG}(T^\ell; Y)$ on a validation set.

Step 2: Choose the estimator.

For $d^\ell \leq 10$: use KSG with $k = \sqrt{n}$ (where n is the batch size). For $d^\ell \in [10, 50]$: use KSG with $k = 5$ and verify stability by checking that the estimate changes by less than 10% when k is doubled. For $d^\ell > 50$: use MINE with a 2-layer MLP critic (100 hidden units, ReLU) and batch size $n \geq 1000$.

Step 3: Add noise injection.

Add Gaussian noise at level $\sigma = 0.1 \cdot \text{std}(T^\ell)$ (10% of the activation standard deviation) to make MI finite and well-defined. Use the same σ throughout training to ensure comparability across time steps.

Step 4: Interpret the results.

Plot $I_\sigma(X;T^\ell)$ and $I_\sigma(T^\ell;Y)$ for each layer ℓ and each training checkpoint. Look for:

- Monotone decrease of both quantities with depth: expected by the DPI.
- Large decrease of $I_\sigma(X;T^\ell)$ between consecutive layers: indicates information loss, potentially due to saturation or dead neurons.
- Small $I_\sigma(T^\ell;Y)$ for early layers: indicates that the early layers have not yet learned task-relevant features.
- Convergence of $I_\sigma(T^L;Y)$ to $H(Y)$: indicates that the network has learned to extract all available information about Y.
- Approach of $I_\sigma(T^L;X)$ to $\log C$: indicates neural collapse (Section 4.5.3).

4.D Representational Similarity and Transfer Learning

Representational similarity analysis (RSA, Kriegeskorte et al. 2008) and its information-theoretic extension provide a principled framework for understanding when and why representations transfer between tasks.

The CKA similarity between two representations T and T' (Definition 4.4) equals 1 if and only if T and T' span the same subspace (up to linear transformation). CKA = 0 if T and T' are statistically independent. For transfer learning, we want the source and target representations to have high CKA: $\text{CKA}(T_{\text{source}}, T_{\text{target}}) \approx 1$ means the source representation is a linear transformation of the target representation, so the source features are directly usable for the target task.

Information-theoretic transfer learning bound: if $\text{CKA}(T^L_{\text{source}}, T^L_{\text{target}}) \geq 1 - \varepsilon$, then the source features can be used for the target task with at most ε additional error per prediction:

Proposition 4.A (CKA and Transfer Error).
Let T_S be the representation learned on the source task and T_T the representation for the target task. If $\text{CKA}(T_S, T_T) \geq 1 - \varepsilon$ and both representations have the same dimension d, then there exists a linear map W such that:
$E[\| W T_S - T_T \|^2] \leq \varepsilon \cdot E[\| T_T \|^2]$.
Consequently, a linear probe trained on top of T_S achieves accuracy $\geq$ accuracy(T_T) $- O(\sqrt{\varepsilon})$ on the target task.

This proposition gives a quantitative bound on the transfer error in terms of CKA: if the source and target representations are nearly identical (CKA $\approx$ 1), the transfer error is small. Empirically, CKA between ResNet-50 features trained on ImageNet and features trained on CIFAR-10 is approximately 0.7–0.8, suggesting moderate transferability —

consistent with the empirical observation that ImageNet features transfer well to CIFAR-10 but not perfectly.

5.A Manifold Unfolding and the Role of Depth

A key geometric operation performed by deep networks is manifold unfolding: transforming a curved, tangled manifold in the input space into a flatter, more linearly separable manifold in the representation space. Information geometry provides the tools to make this precise.

Consider a simple example: the two-spiral dataset, where two interlocked spiral-shaped curves in $\mathbb{R}^2$ must be classified. The data lies on two 1-dimensional manifolds (the spirals) that are interleaved in $\mathbb{R}^2$. A linear classifier cannot separate them. A deep network unfolds the spirals into two disjoint line segments in the representation space, making linear classification trivial.

The unfolding operation corresponds to increasing the length of the manifolds (measured in the Fisher-Rao metric) while separating them in the representation space. From the coarea formula perspective (Theorem 5.2), the unfolding operation changes the fibre structure: in the input space, the fibres of the classification map are single points on each spiral; in the representation space, the fibres are larger (the spirals are mapped to line segments, with multiple input points mapping to nearby representation points on the same segment).

Formally, the unfolding operation can be characterised by the Jacobian of the layer map. Consider a layer $T^\ell = f^\ell(T^{\ell-1})$ where the input manifold has intrinsic dimension d and the output manifold has intrinsic dimension $d' \leq d$. The Jacobian ∇f^ℓ restricted to the tangent space of the input manifold has rank d'. The coarea formula gives:

$$\mathit{Length}(f^\ell(\gamma)) = \int_\gamma J_1 f^\ell(x)\, ds,$$

where γ is a curve on the input manifold and $J_1 f^\ell(x) = \| \nabla f^\ell(x) \|$ is the 1-Jacobian. If $J_1 f^\ell(x) > 1$ on average (the layer stretches the manifold), the unfolding operation increases the manifold length. If $J_1 f^\ell(x) < 1$, the layer contracts the manifold.

ReLU networks have a characteristic piecewise-linear Jacobian: $J_1 f^\ell(x) = \| W^\ell D(x) \|$ where $D(x) = \mathrm{diag}(\mathbb{1}[w_j^T x > 0])$ is the diagonal activation matrix. As the network is trained, the activation patterns $D(x)$ adapt to the data, increasing $J_1 f^\ell$ in directions of high class variation and decreasing it in directions of low class variation.

5.B Curvature, Geodesics, and the Representational Geometry

The geometry of the representation manifold — specifically its curvature — determines how well a linear classifier can separate classes in the representation space. Flat representations (zero curvature) can be perfectly separated by linear classifiers; curved representations require non-linear classifiers.

The Gaussian curvature K of a 2-dimensional representation manifold is related to the Fisher information by the Gauss-Bonnet theorem: $\int_M K \, dA = 2\pi \, \chi(M)$, where $\chi(M)$ is

the Euler characteristic of the manifold. For a sphere ($\chi = 2$), $K > 0$; for a torus ($\chi = 0$), $K = 0$; for a surface with handles ($\chi < 0$), $K < 0$.

For classification tasks with C classes, the ideal representation manifold in R^{log C} has zero curvature (it is flat): the C class-conditional distributions are well-separated Gaussians in the representation space, and a linear classifier can achieve perfect separation. Neural collapse (Section 4.5.3) is the extreme case of flat representation: all within-class variance collapses, and the class means form a simplex ETF that is exactly flat (all pairwise distances equal).

The curvature of intermediate representations can be estimated by measuring the geodesic distance (along the manifold) vs. the ambient distance (in the representation space) between pairs of data points. If the geodesic distance equals the ambient distance (the manifold is flat), then $\| t_i - t \| = d_geo(t_i, t)$ for all pairs. If the manifold is curved, $\| t_i - t \| < d_geo(t_i, t)$ (the manifold curves away from the straight-line path).

5.C Dimension Reduction and Information-Theoretic Guarantees

Principal component analysis (PCA) reduces the dimension of data from D to d by projecting onto the top-d eigenvectors of the data covariance matrix. The information loss in this projection (for Gaussian data) is:

$$I(X; X_PCA) - I(X; X) = -\Sigma_{i=d+1}^{D} (1/2) \log(1 + \sigma_i^2/\sigma_noise^2),$$

where σ_i^2 are the eigenvalues of the data covariance and σ_noise^2 is the observation noise variance. For $\sigma_i \gg \sigma_noise$ (signal-dominated directions), the information loss is small; for $\sigma_i \approx \sigma_noise$ (noise-dominated directions), the information loss is approximately $(1/2) \log 2 \approx 0.5$ nats per discarded direction.

This information-theoretic analysis of PCA provides a principled basis for choosing the reduced dimension d: retain all directions where $\sigma_i > \sigma_noise$ (signal exceeds noise), and discard the rest. The optimal d is determined by the signal-to-noise ratio of the data, not by computational constraints.

For non-Gaussian data, PCA is suboptimal: it only captures linear dependencies, while non-linear dependencies (captured by MI and higher-order statistics) are ignored. Non-linear dimension reduction methods (t-SNE, UMAP, autoencoders) can capture non-linear structure, but they provide no information-theoretic guarantees of the form available for PCA.

Theorem 5.A (PCA Information Loss Bound).
For $X \sim N(0, \Sigma)$ with eigenvalues $\sigma_1^2 \geq \ldots \geq \sigma_D^2$ and PCA projection to dimension d:
$I(X; X_PCA) \geq I(X; Y) - \Sigma_{i=d+1}^{D} I(X_i; Y)$,
where X_i is the projection of X onto the i-th eigenvector. The information loss for predicting Y is bounded by the sum of MI contributions of the discarded directions.

This theorem provides a principled criterion for choosing d in supervised PCA: retain all directions that contribute significantly to I(X;Y). The supervised PCA direction selection

criterion — retain direction i if $I(X_i;Y) > \text{threshold}$ — is information-theoretically optimal for Gaussian data.

6.A Rate-Distortion Theory for Structured Data

Classical rate-distortion theory assumes i.i.d. sources: the data $x^n = (x_1, \ldots, x_n)$ is a sequence of independent draws from $p(x)$. For structured data (images, audio, text), the source is not i.i.d. and the rate-distortion analysis must account for the source's internal structure.

For stationary ergodic sources (including many natural language models), the rate-distortion function per symbol $R(D) = \lim_{n\to\infty} (1/n)\, R_n(D)$ is well-defined and equals the capacity of the rate-distortion problem with distortion measure d averaged over the source. The Shannon-McMillan-Breiman theorem ensures that the rate-distortion function is achieved by typical sequences.

For images with spatial structure, the rate-distortion function can be expressed in terms of the two-dimensional power spectral density $S(\omega_1, \omega_2)$ of the image:

$$R(D) = \iint (1/2)\, \max\{0, \log(S(\omega_1,\omega_2)/\lambda)\}\, d\omega_1\, d\omega_2,$$

where λ is the water-filling level chosen to achieve average distortion D. This is the multi-dimensional extension of the Gaussian rate-distortion function (Theorem 6.3). The water-filling solution allocates more bits to frequencies where the image has high power (low-frequency, large-scale structure) and fewer bits to high-frequency, fine-grained structure.

This analysis explains why JPEG compression (which transforms images to the DCT domain and allocates bits inversely proportional to the frequency) is near-optimal for natural images: the DCT is approximately the optimal transform for images with the $1/f$ power spectral density characteristic of natural scenes.

6.B Normalizing Flows and Exact Density Estimation

Normalizing flows (Dinh et al. 2014; Rezende and Mohamed 2015) construct flexible density models by composing simple invertible transformations. The density of a flow-based model is:

$$\log p_\theta(x) = \log p(z) + \sum_{k=1}^{K} \log \left| \det J_{f_k}(f_{k-1} \circ \ldots \circ f_1(x)) \right|,$$

where $z = f_K \circ \ldots \circ f_1(x)$ is the latent code and each f_k is an invertible transformation with tractable Jacobian determinant. Common choices for f_k include:

- Coupling layers (NICE, RealNVP): split x into (x_A, x_B) and transform $x_B \to x_B \cdot \exp(s(x_A)) + t(x_A)$ where s, t are neural networks. The Jacobian is triangular with diagonal entries $\exp(s(x_A))$, giving $\det J = \exp(\sum_i s_i(x_A))$.
- Autoregressive flows (MAF, IAF): transform x elementwise using an autoregressive model: $x_i \to \sigma(x_i;\, x_{<i})$ where σ is a monotone function of x_i for each context $x_{<i}$. The Jacobian is triangular.

- Continuous normalizing flows (CNF, Chen et al. 2018): transform x continuously via an ODE $dx/dt = f_\theta(x, t)$ from t=0 (data) to t=1 (latent). The log-density change is $\partial \log p_t(x_t)/\partial t = -\mathrm{tr}(\partial f_\theta/\partial x_t)$ (the trace of the Jacobian).

From the rate-distortion perspective, normalizing flows are lossless compressors: because f is invertible, no information is lost in the transformation, and $I(X; Z) = H(X) = H(Z)$. The bits-per-dimension metric $BPD = -(1/d)\, E[\log_2 p_\theta(X)]$ is exactly the description rate (in bits per dimension) of the flow-based model.

The relationship between flow-based models and MDL: a flow-based model p_θ defines a code for data x with code length $-\log_2 p_\theta(x)$. Minimising the negative log-likelihood (training the flow) is equivalent to finding the model that achieves the shortest expected code length — exactly the MDL principle. The optimal flow (achieving the true data density p(x)) gives expected code length H(X) = entropy rate of the data, which is the fundamental limit.

6.C Diffusion Models: A Unified Rate-Distortion View

Diffusion models can be understood as multi-scale rate-distortion coders that progressively reveal the structure of the data distribution at different noise scales. This perspective, developed by Kingma et al. (2021), provides a unified theoretical framework for understanding diffusion model training and inference.

The signal-to-noise ratio (SNR) at noise scale t is α^2_t/σ^2_t, where $\alpha_t = e^{-\int_0^t \beta(s)ds/2}$ and $\sigma^2_t = 1 - \alpha^2_t$. The SNR decreases monotonically from $SNR(0) = \infty$ (pure signal) to $SNR(T) \approx 0$ (pure noise).

The bit allocation across noise scales is determined by the Shannon capacity of each 'channel' (the forward diffusion process at scale t): the number of bits that can be reliably transmitted about x_0 through the noisy observation x_t is:

$$C(t) = (1/2)\log(1 + SNR(t)) = (1/2)\log(1 + a^2_t/\sigma^2_t).$$

The total information capacity is $\int_0^T C(t)\, dt \approx \int_0^\infty (1/2)\log(1 + \alpha^2_t/\sigma^2_t)\, dt = H(X)/d$ (the differential entropy per dimension), establishing that the diffusion process transmits exactly H(X)/d bits of information about the data across all noise scales.

The optimal noise schedule minimises the total training loss by allocating model capacity optimally across noise scales. Scales with high SNR (small t) carry fine-grained information about x_0 (sharp edges, textures) and require more model capacity. Scales with low SNR (large t) carry coarse information about x_0 (global structure, semantic content) and require less model capacity.

This analysis predicts that the optimal noise schedule $\beta^*(t)$ should have more time steps at small t (where the SNR is high and information is fine-grained) and fewer steps at large t (where the SNR is low). The cosine noise schedule (Nichol and Dhariwal 2021) approximately achieves this by spending more time in the low-noise regime (small t) compared to the linear schedule.

7.A The Fundamental Limits of Learning: A Unified View

The four lower bound techniques developed in Chapter 7 (Fano, Le Cam, Assouad, IZS) are all manifestations of a single principle: learning requires information, and the amount of information available from n samples is finite. Each technique makes this principle precise in a different way.

Fano's inequality relates the error probability to the mutual information between the parameter M and the observations Y. The key step is the data processing inequality: I(M;Y) ≥ I(M;M̂(Y)), so the mutual information between M and the observations bounds the accuracy of any estimator.

Le Cam's method is Fano's inequality for two hypotheses: the minimum of the two error probabilities equals (1 − ‖ P₀ − P₁ ‖_TV)/2, which is controlled by D_KL(P₀ ‖ P₁) via the bound TV ≤ √(D_KL/2).

Assouad's lemma is Fano's inequality for a product of two-hypothesis problems: the d-dimensional parameter space {±1}^d decomposes into d independent binary problems, each with error probability controlled by the marginal KL divergence D_KL(P^{(j)}_+ ‖ P^{(j)}_-).

The IZS method (Theorem 7.5) is Fano's inequality applied to the algorithm A itself: the mutual information I(A(S);S) between the algorithm's output and the training set bounds the generalisation gap, because the generalisation gap depends on how much information the algorithm has extracted from S.

Theorem 7.A (Unified Information-Theoretic Lower Bound).
All four lower bound techniques are special cases of the following master theorem: for any estimator θ̂(Y) of θ from observations Y ~ P_θ, and any distinguishability measure D between the hypotheses:
Risk(θ̂) ≥ Ω(sep(Θ) · (1 − √(n · D(P₀, P₁)/2))),
where sep(Θ) is the separation between hypotheses in the estimation metric, D(P₀,P₁) is the per-sample distinguishability (KL divergence for Le Cam, TV for Assouad, MI for Fano), and n is the number of observations.

The key insight is that all lower bounds reduce to the question: how distinguishable are the best and worst hypotheses from n observations? If they are indistinguishable (D → 0), the estimator must have large risk. If they are highly distinguishable (D → ∞ with n), the estimator can achieve low risk.

7.B Minimax Lower Bounds for Specific Deep Learning Tasks

We derive minimax lower bounds for several specific deep learning tasks using the techniques of Chapter 7.

Image classification with label noise.
Consider classifying images X ∈ R^D into C classes with label noise rate η: each label Y is flipped to a random incorrect class with probability η. The minimax risk for n samples with feature dimension D and C classes is:

$$R^* = \Omega(\sqrt{(C \log C \,/\, (n(1-2\eta)^2))}).$$

This lower bound follows from Fano's inequality applied to the C-class problem, with the effective sample size reduced by $(1-2\eta)^2$ due to label noise. For $\eta = 0.5$ (random labels), the denominator vanishes and the lower bound is vacuous — consistent with the fact that no classifier can do better than random with completely random labels.

Next-token prediction.

For autoregressive language modelling over a vocabulary of V tokens with sequence length T, the minimax cross-entropy per token is:

$$L^* = \Omega(h_\infty + \sqrt{(V \log V \,/\, (nT))}),$$

where h_∞ is the entropy rate of the language (the irreducible error). The first term is the information-theoretic limit; the second term is the statistical lower bound that decreases as $1/\sqrt{(nT)}$ — consistent with the observed scaling law $L \propto n^{-0.095}$ for large language models.

Policy learning in reinforcement learning.

For learning a policy π in a Markov decision process with S states, A actions, and horizon H, the minimax sample complexity (number of trajectories) is:

$$n^* = \Omega(S A H \,/\, \varepsilon^2),$$

where ε is the target suboptimality. This lower bound (derived via Fano's inequality on the set of MDPs) is tight: model-based RL algorithms (UCRL, RMAX) achieve this rate up to logarithmic factors.

7.C The Sample Complexity of Fine-Tuning Foundation Models

Foundation models (BERT, GPT, CLIP) are pre-trained on massive datasets and fine-tuned on downstream tasks with much smaller datasets. The information-theoretic analysis of fine-tuning reveals the conditions under which pre-training is beneficial and the minimum number of fine-tuning samples needed.

Let θ^* be the optimal parameters for the downstream task (unknown) and θ_{pre} be the pre-trained parameters (known). Fine-tuning adapts θ_{pre} to θ^* using n downstream samples. The key parameter is $\Delta = \| \theta^* - \theta_{pre} \|$ — the 'adaptation distance' between the pre-trained and optimal parameters.

Theorem 7.B (Sample Complexity of Fine-Tuning).

For fine-tuning a p-parameter model from pre-trained parameters θ_{pre} to optimal parameters θ^* using n downstream samples, the minimum number of samples to achieve generalisation gap $\leq \varepsilon$ with probability $\geq 1 - \delta$ is:

$n^* = \Omega(\Delta^2 \,/\, (\varepsilon^2 \cdot \sigma^2_0))$,

where σ^2_0 is the prior variance (weight decay). Fine-tuning requires fewer samples than training from scratch (which requires $n^* = \Omega(p/\varepsilon^2)$) by a factor of $p \,/\, (\Delta^2/\sigma^2_0)$.

For LoRA (Low-Rank Adaptation) with rank r, the adaptation distance satisfies $\Delta_{LoRA} \leq \Delta_{full} \cdot \sqrt{(r/p)}$, so the sample complexity is further reduced by r/p.

This theorem explains why fine-tuning with LoRA (Hu et al. 2022) is data-efficient: by constraining the adaptation to a low-rank subspace, LoRA reduces the effective adaptation distance Δ by a factor of $\sqrt{(r/p)}$, requiring r/p fewer samples compared to full fine-tuning. For $r = 8$ and $p = 10^9$ (typical for large language model LoRA), the sample complexity is reduced by a factor of 10^{-7} compared to training from scratch.

The information-theoretic perspective on few-shot learning: in k-shot learning (k examples per class), the total number of samples is $n = k \cdot C$. The minimum number of shots per class to achieve ε-accuracy with C classes is:

$$k^* = \Omega(\Delta^2_{class} / (\varepsilon^2 \cdot \sigma^2_0 \cdot C)),$$

where Δ_{class} is the adaptation distance for a single class. For the ImageNet pre-trained model and CIFAR-100 downstream task, Δ_{class} is small (the classes are similar between the two datasets), giving $k^* = O(1)$ — consistent with the empirical observation that 5-shot learning achieves near-full fine-tuning accuracy for this transfer pair.

8.A The Gibbs Algorithm in Practice: MCMC and Langevin Dynamics

The Gibbs algorithm $A^*(S) \propto P(h)\, e^{-\beta n \hat{L}(h)}$ is the theoretically optimal learning algorithm, but exact sampling from this distribution is generally intractable. In practice, approximate sampling methods are used. The most important are Markov Chain Monte Carlo (MCMC) and Stochastic Gradient Langevin Dynamics (SGLD).

MCMC for Bayesian neural networks: the Metropolis-Hastings algorithm generates a Markov chain $\{w_t\}_{t=0}^T$ that converges to the Gibbs posterior $Q^*_\beta \propto P\, e^{-\beta n \hat{L}}$. At each step, a proposal w' is generated from a proposal distribution $q(w'|w_t)$, and the new sample is accepted with probability $\min(1, P(w')e^{-\beta n \hat{L}(w')}\, q(w_t|w') / (P(w_t)\, e^{-\beta n \hat{L}(w_t)}\, q(w'|w_t)))$. This ensures detailed balance and convergence to the Gibbs posterior.

MCMC for neural networks has two main challenges: (1) the high dimensionality ($p = 10^7$ — 10^9 parameters) makes the state space enormous, requiring many steps to mix; and (2) the loss landscape is highly non-convex, with many local minima separated by high barriers. Standard MCMC methods (Metropolis-Hastings, Gibbs sampling) mix extremely slowly in this regime.

Stochastic Gradient Langevin Dynamics (SGLD, Welling and Teh 2011): combines SGD with Langevin dynamics to sample approximately from the Gibbs posterior. The SGLD update is:

$$w_{t+1} = w_t - (\eta_t/2)\, \nabla_w \hat{L}(w_t) + \sqrt{\eta_t} \cdot \varepsilon_t, \quad \varepsilon_t \sim N(0, I),$$

where η_t is a decaying learning rate schedule. For $\eta_t \to 0$ and $\Sigma_t \eta_t = \infty$ (e.g., $\eta_t = \eta_0/t$), SGLD converges to the correct Gibbs posterior (Teh et al. 2016). For constant $\eta_t = \eta$ (the standard SGD learning rate), SGLD samples from an approximate Gibbs posterior with effective temperature $T_{eff} \propto \eta$.

The information-theoretic interpretation of SGLD: each step of SGLD takes a gradient step (increasing $\hat{L}$ information) and adds noise (increasing entropy). The equilibrium is

reached when the information gain from the gradient step equals the information loss from the noise, which occurs when the weight distribution Q_t matches the Gibbs posterior Q*_β at inverse temperature $\beta = 1/\eta$.

Cyclical SGLD (Zhang et al. 2020): uses a cyclical learning rate schedule that alternates between large learning rates (exploration, high temperature) and small learning rates (exploitation, low temperature). This allows SGLD to explore multiple modes of the posterior (at high temperature) while sampling accurately within each mode (at low temperature). Cyclical SGLD provides better uncertainty estimates than standard SGLD and can improve ensemble diversity.

8.B The Free Energy Perspective on Learning

The Gibbs algorithm has a natural interpretation in terms of statistical mechanics: it minimises the free energy $F(Q) = E_Q[\hat{L}(h)] - (1/\beta)\, H(Q)$, where $H(Q) = -E_Q[\log Q(h)]$ is the entropy of the posterior distribution Q.

The free energy F(Q) is related to the PAC-Bayes bound by $F(Q) = E_Q[\hat{L}(h)] + (1/\beta)\, D_{KL}(Q \,/\!/\, P) - (1/\beta)\, H(P)$ (up to a constant). Minimising F(Q) over Q is equivalent to minimising the PAC-Bayes upper bound on L(Q) at inverse temperature β.

The free energy perspective has several important consequences:

- Temperature annealing: as β increases (cooling), the posterior Q concentrates around the minimum of $\hat{L}(h)$. Simulated annealing uses a schedule $\beta_t \to \infty$ to find the global minimum of $\hat{L}$, but the optimal (PAC-Bayes-optimal) β is finite — infinite cooling gives the MAP estimator, which is not the Gibbs posterior.
- The entropy bonus: the term $-H(Q)/\beta$ in F(Q) penalises overly concentrated posteriors. A posterior that assigns all its mass to a single hypothesis (the MAP estimator) has $H(Q) = 0$ and pays the full cost of the training loss without any entropy bonus. A diffuse posterior (high H(Q)) can tolerate some increase in training loss in exchange for a larger entropy bonus, leading to flatter minima.
- Relation to stochastic weight averaging: SWA computes the average $w_{SWA} = (1/T)\, \Sigma_t\, w_t$ of the weight iterates, which approximates the mean of the Gibbs posterior. The SWA predictor $f_{w_{SWA}}(x) \approx E_Q[f_w(x)]$ is the Bayesian model average under Q, which has lower generalisation error than any individual w_t by Jensen's inequality.

8.C Information-Theoretic Analysis of Prompt Engineering

Prompt engineering — the practice of designing input prompts that elicit better responses from large language models — can be analysed from an information-theoretic perspective. A prompt P is a sequence of tokens that precedes the user's query Q, and the model generates a response R conditioned on (P, Q).

The information content of a prompt: a good prompt increases $I(R; Y \mid Q)$ — the mutual information between the response R and the correct answer Y, given the query

Q. A bad prompt decreases I(R; Y | Q) by introducing irrelevant information or incorrect assumptions.

Chain-of-thought (CoT) prompting (Wei et al. 2022) asks the model to generate a reasoning chain before the final answer. From an information-theoretic perspective, CoT increases the effective context length available for generating the answer: instead of computing I(R; Y | Q) directly, the model computes I(CoT; Y | Q) + I(R; Y | Q, CoT). If the reasoning chain CoT is informative about Y, the total MI is larger than without CoT.

Proposition 8.A (CoT and Information Decomposition).
Let R = (CoT, A) where CoT is the reasoning chain and A is the final answer. Then:
$I(R; Y \mid Q) = I(CoT; Y \mid Q) + I(A; Y \mid Q, CoT)$.
CoT is beneficial (increases $I(R;Y|Q)$) if and only if $I(CoT; Y \mid Q) > 0$, i.e., the reasoning chain carries information about the correct answer beyond what the query alone provides.

This decomposition justifies CoT prompting information-theoretically: the reasoning chain acts as a 'bottleneck' variable that captures intermediate information about Y, allowing the final answer A to be more accurately conditioned on Y through the chain of reasoning.

The information gain from few-shot examples: in k-shot prompting, the prompt contains k (query, answer) pairs followed by the test query. The information gain from the k examples is:

$$\Delta I_k = I(A; Y \mid Q, examples_{1:k}) - I(A; Y \mid Q) \geq 0.$$

This information gain decreases as k increases (diminishing returns) and saturates at I(X;Y) (the mutual information between the query features and the correct answer). The optimal k balances the information gain against the context length cost of the prompt.

Appendix A

Worked Examples in Information Theory for Deep Learning

This appendix provides worked examples that illustrate the main theorems of this volume. Each example is designed to be self-contained and to show how the theoretical results translate into concrete calculations.

A.1 Computing the IB Curve for a Simple Binary Source

Problem: compute the IB curve for X uniform on {0,1,2,3} (4 symbols, 2 bits) and Y = X mod 2 (binary target, 1 bit). The joint distribution is p(x,y) = 1/4 for all (x,y) with y = x mod 2.

Step 1: compute the relevant information quantities.

$$H(X) = \log 4 = 2 \text{ bits}, \quad H(Y) = \log 2 = 1 \text{ bit}, \quad I(X;Y) = H(Y) - H(Y|X) = 1 - 0 = 1 \text{ bit}.$$

The maximum achievable I(T;Y) is I(X;Y) = 1 bit (achieved by a sufficient statistic T = X mod 2).

Step 2: find the IB curve by varying β.

For $\beta < \beta_1 = 1$ (first critical value): the optimal encoder is $p^*(t|x) = 1/2$ for all x, $t \in \{0,1\}$. This gives I(X;T) = 0 and I(T;Y) = 0 — the trivial solution.

For $\beta > \beta_1 = 1$: the optimal encoder is the binary partition T = X mod 2 (T=0 for $X \in \{0,2\}$, T=1 for $X \in \{1,3\}$). This gives I(X;T) = H(T) = 1 bit and I(T;Y) = I(X;Y) = 1 bit — the sufficient statistic.

The IB curve has only two points: (0,0) for $\beta < 1$ and (1,1) for $\beta > 1$, with a phase transition at $\beta = 1$. The transition occurs because the single most informative partition (T = X mod 2) achieves $I(T;Y)/I(X;T) = 1 = \beta_1$.

Verification using the self-consistent equations: at $\beta = 2$, the Gibbs encoder is:

$$p^*(t=0|x) \propto p(t=0) \exp(-2\, D_{KL}(p(y|x) \,\|\, p(y|t=0))),$$

For x=0 (y=0 deterministically), $D_{KL}(p(y|x=0) \,\|\, p(y|t=0)) = D_{KL}(\delta_0 \,\|\, \text{Bernoulli}(1/2)) = \log 2$. For x=2 (y=0 deterministically), same. For x=1 (y=1 deterministically), $D_{KL}(p(y|x=1) \,\|\, p(y|t=0)) = D_{KL}(\delta_1 \,\|\, \text{Bernoulli}(1/2)) = \log 2$. So all D_{KL} values are equal, and the encoder is uniform — but this contradicts the expected solution T = X mod 2. We need to initialise closer to the true solution; the self-consistent equations have multiple fixed points.

Starting from $p^*(t=0|x) = 1$ for $x \in \{0,2\}$ and $p^*(t=0|x) = 0$ for $x \in \{1,3\}$: $p(t=0) = 1/2$. $D_{KL}(p(y|x=0) \,\|\, p(y|t=0)) = D_{KL}(\delta_0 \,\|\, \delta_0) = 0$ (since $p(y=0|t=0) = 1$). $D_{KL}(p(y|x=1) \,\|\, p(y|t=0)) = D_{KL}(\delta_1 \,\|\, \delta_0) = \infty$. So $p^*(t=0|x=0) \propto (1/2)\, e^0 = 1/2$ and $p^*(t=0|x=1) \propto (1/2)\, e^{\{-\infty\}} = 0$. This confirms the IB-optimal encoder is T = X mod 2.

A.2 PAC-Bayes Bound Computation: MNIST Example

Problem: compute the PAC-Bayes bound for a small fully-connected network (784→100→10) trained on MNIST (n = 60,000 training examples) with the following setup:

- Prior $P = N(0, \sigma_0^2 I)$ with $\sigma_0 = 0.03$ (he initialisation standard deviation).
- Posterior $Q = N(w^*, \sigma^2 I)$ with $\sigma = \sigma_0$ (posterior variance = prior variance).
- Trained weights: $\| w^* \|_2 = 45$ (typical for a network of this size with weight decay).
- Number of parameters: $p = 784 \times 100 + 100 \times 10 = 79{,}400$.
- Training loss: $\hat{L}(Q) = 0.05$ (5% training error).
- Confidence level: $\delta = 0.05$.

Step 1: compute the KL divergence.

$$D_{KL}(Q \| P) = \| w^* \|^2 / (2\sigma_0^2) = 45^2 / (2 \times 0.03^2) = 2025 / 0.0018 = 1{,}125{,}000.$$

Step 2: apply the McAllester-Seeger bound.

$$L(Q) \le \hat{L}(Q) + \sqrt{[(D_{KL}(Q \| P) + \log(2\sqrt{n}/\delta))/(2n)]}$$

$$= 0.05 + \sqrt{[(1{,}125{,}000 + \log(2\sqrt{60000}/0.05))/(2 \times 60000)]}$$

$$= 0.05 + \sqrt{[(1{,}125{,}000 + \log(9798))/(120{,}000)]}$$

$$= 0.05 + \sqrt{[(1{,}125{,}000 + 9.19)/120{,}000]}$$

$$= 0.05 + \sqrt{[9.375]} = 0.05 + 3.06 = 3.11.$$

This bound is vacuous (> 1 in classification). The problem is that $\| w^* \|_2 = 45$ is too large relative to $\sigma_0 = 0.03$.

Step 3: optimise the posterior variance σ. Setting $\sigma = \sigma^*$, where σ^* minimises $D_{KL}(Q \| P) = p(\sigma/\sigma_0 - 1 - \log(\sigma/\sigma_0))/2 + \| w^* \|^2/(2\sigma_0^2)$ subject to $\sigma \le \sigma_0$. The minimum is at $\sigma = \sigma_0$ (which we already used) — the bound is dominated by $\| w^* \|^2/(2\sigma_0^2)$. We need to either reduce $\| w^* \|$ (by using stronger weight decay) or increase σ_0 (by using a looser prior).

Step 4: use a data-dependent prior. By sample splitting (Theorem 2.6) with m = 5000 samples for the prior and n-m = 55,000 for the bound: run a short training on 5000 samples to get w_init (with $\| w_{init} \| \approx 5$). Set $P_{data} = N(w_{init}, \sigma_0^2 I)$. Now $D_{KL}(Q \| P_{data}) = \| w^* - w_{init} \|^2/(2\sigma_0^2) \approx (45\text{-}5)^2/0.0018 \approx 888{,}889$ — still large.

Step 5: use the optimised PAC-Bayes approach of Dziugaite and Roy. Directly minimise the PAC-Bayes objective over σ^* by treating it as a hyperparameter. Setting $\sigma^* = \sqrt{(\| w^* \|^2/(2p \log(2\sqrt{n}/\delta)))} \approx \sqrt{(2025/(2 \times 79400 \times 9.19))} \approx \sqrt{(0.00139)} \approx 0.037$. Then $D_{KL} \approx p(0.037/0.03 - 1 - \log(0.037/0.03))/2 + 2025/0.0018 \approx 1{,}125{,}000$. The bound remains vacuous because $\| w^* \|$ is too large relative to σ_0.

Lesson: for standard neural network training, PAC-Bayes bounds require either (a) strong weight decay (reducing $\|w^*\|$ to $O(\sqrt{(n/p)})$), or (b) data-dependent priors that are very close to the posterior. The Dziugaite-Roy approach achieves non-vacuous bounds by directly optimising the PAC-Bayes objective rather than computing a post-hoc bound on a conventionally trained network.

A.3 MDL Model Selection: Linear Regression Example

Problem: select between polynomial models of degree 1, 2, 3 for fitting $n = 100$ noisy observations from the function $f(x) = x^2 + 0.1 \sin(10x) + \varepsilon$, $\varepsilon \sim N(0, 0.1^2)$.

True model order: degree 2 (quadratic). However, the high-frequency oscillation $\sin(10x)$ is not captured by any polynomial of degree ≤ 5 without very high degree.

Data: 100 observations $x_i \sim U[0,1]$, $y_i = f(x_i) + \varepsilon_i$.

Step 1: compute the MLE log-likelihood for each degree.

$$\log p(y^n \mid \hat{\theta}_{deg}, deg) = -(n/2) \log(2\pi \hat{\sigma}^2_{deg}) - RSS_{deg}/(2\hat{\sigma}^2_{deg}),$$

where $RSS_{deg} = \|y - X_{deg} \hat{\theta}_{deg}\|^2$ is the residual sum of squares and $\hat{\sigma}^2_{deg} = RSS_{deg}/n$.

Numerical results (typical values):

- Degree 1: RSS = 0.85, $\hat{\sigma}^2 = 0.0085$, $\log p = -100/2 \log(0.0534) - 50 \approx 136$.
- Degree 2: RSS = 0.11, $\hat{\sigma}^2 = 0.0011$, $\log p = -100/2 \log(0.00689) - 50 \approx 229$.
- Degree 3: RSS = 0.10, $\hat{\sigma}^2 = 0.001$, $\log p = -100/2 \log(0.00629) - 50 \approx 231$.

Step 2: apply BIC (first-order MDL). $BIC = -\log p + (d/2) \log n$:

- Degree 1 (d=2): $BIC = -136 + 1 \times \log(100) \approx -136 + 4.6 = -131.4$.
- Degree 2 (d=3): $BIC = -229 + 1.5 \times \log(100) \approx -229 + 6.9 = -222.1$.
- Degree 3 (d=4): $BIC = -231 + 2 \times \log(100) \approx -231 + 9.2 = -221.8$.

BIC selects degree 3 (smallest BIC = −221.8 is largest negative value... wait, BIC is typically minimised). Correcting: BIC penalises $-\log p$ (larger is better, so we minimise $-\log p$ + penalty):

- Degree 1: $BIC = 136 \times (-1) + 4.6 = -131.4$? No — $BIC = -2 \log p + d \log n$.
- Degree 1: $BIC = -2 \times 136 + 2 \times 4.6 = -272 + 9.2 = -262.8$. Smaller = better.
- Degree 2: $BIC = -2 \times 229 + 3 \times 4.6 = -458 + 13.8 = -444.2$.
- Degree 3: $BIC = -2 \times 231 + 4 \times 4.6 = -462 + 18.4 = -443.6$.

BIC selects degree 2 (smallest BIC = −444.2)! Correct — the BIC penalty of $3 \times 4.6 = 13.8$ for degree 2 vs $4 \times 4.6 = 18.4$ for degree 3 outweighs the small log-likelihood improvement of degree 3 over degree 2.

This example illustrates MDL model selection: the correct model (degree 2) is selected because the MDL penalty for degree 3 ($\log C_n(\text{poly3})$) outweighs the marginal improvement in fit from adding a cubic term.

A.4 The MINE Estimator: A Numerical Example

Problem: estimate $I(X;T)$ for $X \sim N(0,1)$ and $T = X + \varepsilon$ where $\varepsilon \sim N(0,\sigma^2)$. The true MI is $I(X;T) = (1/2)\log(1 + 1/\sigma^2)$ (Gaussian channel capacity).

Setup: generate $n = 10{,}000$ paired samples (x_i, t_i) and $n = 10{,}000$ marginal samples $\tilde{t}_j$ (drawn from the marginal distribution of T, which is $N(0, 1+\sigma^2)$). Train a critic network $f_\phi(x,t)$ to maximise $\hat{I}_{MINE}$.

Critic architecture: 2-layer MLP with 100 hidden units, ReLU activation, input dimension 2. Optimise with Adam, learning rate 10^{-3}, batch size 256, for 10,000 steps.

Results for $\sigma = 0.5$ (true $I(X;T) = (1/2)\log(1 + 4) = (1/2)\log 5 \approx 0.805$ nats):

- MINE estimate after 1,000 steps: 0.72 (10% underestimate due to Jensen's bias).
- MINE estimate after 10,000 steps: 0.79 (2% underestimate, approaching true value).
- MINE-f estimate after 1,000 steps: 0.79 (1% underestimate, lower variance).
- KSG estimate (k=5): 0.800 ± 0.003 (consistent, minimal bias for 1D case).

For $\sigma = 0.1$ (true $I(X;T) = (1/2)\log(101) \approx 2.31$ nats):

- MINE estimate after 10,000 steps: 2.10 (9% underestimate, Jensen's bias larger for high MI).
- MINE-f: 2.25 (3% underestimate).
- KSG: 2.30 ± 0.01 (consistent, still reliable for 1D).

For $\sigma = 0.01$ (true $I(X;T) \approx 4.61$ nats):

- MINE: 3.8 (18% underestimate — large bias at very high MI).
- MINE-f: 4.3 (7% underestimate).
- KSG: 4.58 ± 0.02 (still accurate for 1D).

Lesson: for high MI ($I > 2$ nats), MINE underestimates significantly due to Jensen's bias. Use MINE-f or KSG for accurate high-MI estimation. For very high MI ($I > 4$ nats), even KSG may struggle due to the k-NN radius becoming very small.

Appendix B

Mathematical Background

This appendix collects mathematical prerequisites for the volume. We assume familiarity with probability theory, linear algebra, and real analysis at the graduate level. The appendix covers measure theory, functional analysis, and convex analysis at the level needed for the main text.

B.1 Measure Theory and Probability

A probability space is a triple $(\Omega, \mathcal{F}, P)$ where Ω is the sample space, $\mathcal{F}$ is a σ-algebra of events, and P is a probability measure on $(\Omega, \mathcal{F})$. A random variable $X : \Omega \to \mathcal{X}$ is a measurable function from Ω to a measurable space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$.

The Radon-Nikodym theorem: if $P \ll Q$ (P is absolutely continuous with respect to Q), then there exists a measurable function $dP/dQ : \mathcal{X} \to [0,\infty)$ (the Radon-Nikodym derivative) such that $P(A) = \int_A (dP/dQ)\, dQ$ for all $A \in \mathcal{B}(\mathcal{X})$. The KL divergence is $D_{KL}(P \,/\!/\, Q) = E_P[\log(dP/dQ)] = \int (dP/dQ) \log(dP/dQ)\, dQ$.

Fubini's theorem: for a jointly measurable function $f(x,y)$ on a product space $(\mathcal{X}\times\mathcal{Y}, \mathcal{B}(\mathcal{X})\otimes\mathcal{B}(\mathcal{Y}), P\times Q)$, if $\iint |f(x,y)|\, dP(x)\, dQ(y) < \infty$, then $\iint f\, dP\, dQ = \int\left(\int f(x,y)\, dP(x)\right) dQ(y) = \int\left(\int f(x,y)\, dQ(y)\right) dP(x)$. Fubini's theorem justifies interchanging the order of integration, which is used repeatedly in the derivations of this volume.

Dominated convergence theorem: if $X_n \to X$ almost surely and $|X_n| \le Y$ a.s. for some integrable Y ($\int Y\, dP < \infty$), then $E[X_n] \to E[X]$. This theorem justifies differentiating under the integral sign (by applying it to the sequence of difference quotients) and is the basis for proving the continuity and differentiability of MI as a function of the distribution parameters.

B.2 Convex Analysis and Duality

A function $f : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ is convex if for all $x, y \in \mathcal{X}$ and $\lambda \in [0,1]$: $f(\lambda x + (1-\lambda)y) \le \lambda f(x) + (1-\lambda) f(y)$. Strict convexity holds when the inequality is strict for $\lambda \in (0,1)$ and $x \neq y$. A twice-differentiable function is convex if and only if its Hessian is positive semidefinite.

Jensen's inequality: for a convex function ϕ and a random variable X with $E[X]$ finite: $\phi(E[X]) \le E[\phi(X)]$. Equality holds if ϕ is linear on the support of X. Jensen's inequality is used pervasively in information theory: it gives $H(X) \ge 0$ (from $\phi = -\log$), $D_{KL}(P \,/\!/\, Q) \ge 0$ (from the convexity of $x \log x$), and many variational bounds.

Legendre-Fenchel transform: for a convex function $f : \mathcal{X} \to \mathbb{R}$, the convex conjugate is $f^*(y) = \sup_{x\in\mathcal{X}} \{\langle y,x\rangle - f(x)\}$. The Donsker-Varadhan representation $D_{KL}(P \,/\!/\, Q) = \sup_f \{E_P[f] - \log E_Q[e^f]\}$ is the Legendre-Fenchel transform of the log-partition function $\log E_Q[e^f]$ as a function of f. This duality is the key to deriving the MINE estimator and the PAC-Bayes change-of-measure lemma.

Minimax theorem (von Neumann, 1928): for a compact convex set X, a compact convex set Y, and a continuous function $f : X \times Y \to R$ that is convex in x and concave in y: $\min_x \max_y f(x,y) = \max_y \min_x f(x,y)$. The minimax theorem is used in the proof of Rissanen's minimax regret result (Theorem 3.1) and in the derivation of the rate-distortion function.

B.3 Information Theory: Key Results

Shannon's source coding theorem: the minimum expected code length for a source X with distribution P over a finite alphabet X is H(X) bits, achieved by the Huffman code. Any code has expected length $\geq H(X)$.

Shannon's channel coding theorem: for a channel with capacity $C = \max_{p(x)} I(X;Y)$, any rate $R < C$ is achievable (there exist codes with rate R and error probability $\to 0$ as code length $\to \infty$). Any rate $R > C$ is not achievable (error probability $\to 1$ for any sequence of codes).

Shannon's rate-distortion theorem: for a source X with distribution P and distortion measure d, the minimum rate $R(D) = \min_{p(\hat{x}|x): E[d(X,\hat{X})] \leq D} I(X;\hat{X})$ is achievable. Any rate $R < R(D)$ cannot achieve distortion D.

The data processing inequality: for a Markov chain $X \to T \to Y$: $I(X;T) \geq I(X;Y)$ and $I(X;Y) \geq I(T;Y)$. Equality holds iff T is a sufficient statistic for X about Y.

Fano's inequality: for any estimator $\hat{\theta}(Y)$ of θ and Markov chain $\theta \to Y \to \hat{\theta}$: $H(\theta|\hat{\theta}) \leq 1 + P(\hat{\theta} \neq \theta) \log |\Theta|$ (the conditional entropy is bounded by the error probability times the log of the alphabet size).

Appendix C

Notation and Glossary

This appendix summarises the notation and key terms used throughout the volume.

C.1 Probability and Information Theory Notation

Random variables are denoted by uppercase letters (X, Y, Z, T, W), their realisations by lowercase (x, y, z, t, w), and their alphabet (support) by calligraphic letters (X, Y, Z).

H(X): Shannon entropy of X. H(X|Y): conditional entropy of X given Y. I(X;Y): mutual information between X and Y. I(X;Y|Z): conditional mutual information. D_KL(P ∥ Q): Kullback-Leibler divergence from Q to P. D_α(P ∥ Q): Rényi divergence of order α. h(X): differential entropy (for continuous X).

p(x): probability mass or density function of X. p(x|y): conditional distribution of X given Y. P_n = p^n: product distribution for n i.i.d. samples. p*(t|x): IB-optimal encoder. Q*_λ: Gibbs posterior at inverse temperature λ. p_NML: normalised maximum likelihood distribution.

C.2 Linear Algebra Notation

∥ v ∥ ₂: Euclidean (ℓ_2) norm of vector v. ∥ v ∥ ₁: ℓ_1 norm. ∥ A ∥ _F: Frobenius norm of matrix A. ∥ A ∥ _σ: spectral norm (largest singular value). tr(A): trace. det(A): determinant. A^T: transpose. A^{-1}: inverse. I_p: p×p identity matrix. diag(v): diagonal matrix with diagonal v.

$\sigma_i(A)$: i-th singular value of A (decreasing order). $\lambda_i(A)$: i-th eigenvalue. SVD_r(A): best rank-r approximation to A. ∇_w f: gradient of f with respect to w. ∇²_w f: Hessian. J_f(x): Jacobian of f at x.

C.3 Glossary of Key Terms

Information Bottleneck (IB):
The variational framework for finding compressed representations T of X that maximally retain information about Y, parametrised by the trade-off parameter β.

PAC-Bayes bound:
A generalisation bound that controls the population risk L(Q) of a distribution Q over hypotheses in terms of D_KL(Q ∥ P) and the empirical risk L̂(Q).

Minimum Description Length (MDL):
The principle that the optimal model is the one that minimises the total description length of the model and the data given the model.

Gibbs algorithm:

The information-theoretically optimal learning algorithm $A^*(S) \propto P\, e^{-\beta nL}$, which simultaneously minimises IZS MI, PAC-Bayes KL, MDL complexity, and IB functional.

Jeffreys prior:
The prior $P^*(w) \propto \sqrt{\det I_F(w)}$ that is simultaneously optimal for PAC-Bayes, MDL, and IB.

Neural Tangent Kernel (NTK):
The kernel $K(x,x') = \nabla_w f_{w_0}(x) \cdot \nabla_w f_{w_0}(x')$ that governs infinite-width network training dynamics.

Neural collapse:
The terminal-phase training phenomenon where last-layer features collapse to class means forming a simplex ETF, achieving $I(T^L;X) \approx \log C$.

Noise-regularised MI:
$I_\sigma(X;T) = I(X; T+\varepsilon)$ for $\varepsilon \sim N(0,\sigma^2 I)$, which is finite for deterministic networks and estimates the Hausdorff dimension of the representation support.

Blahut-Arimoto (BA):
The iterative algorithm for computing the rate-distortion function (and IB curve) for discrete sources, which VIB training generalises to continuous sources.

Differential privacy (DP):
A privacy guarantee that bounds the log-likelihood ratio of mechanism outputs on adjacent datasets by ε.

Rényi DP (RDP):
An extension of DP using Rényi divergence, which provides tight composition bounds.

Hausdorff dimension:
$\dim_H(E) = \inf\{s : H^s(E) = 0\}$, the dimension of a set E in geometric measure theory.

Wasserstein distance:
$W_p(\mu,\nu) = (\inf_{\gamma \in \Pi(\mu,\nu)} \int \| x-y \|^p \, d\gamma)^{1/p}$, the optimal transport distance between probability measures.

CKA (Centred Kernel Alignment):
$CKA(K,L) = \| L^T K \|^2_F / (\| K \|_F \| L \|_F)$, a representational similarity metric invariant to orthogonal transformations and isotropic scaling.

MINE:
Mutual Information Neural Estimation — a neural lower bound on MI using the Donsker-Varadhan representation.

KSG estimator:
The k-nearest-neighbour MI estimator of Kraskov, Stögbauer, and Grassberger (2004), consistent for smooth distributions.

V-usable information:

$I_V(X \to Y) = H(Y) - \inf_{v \in V} E[-\log v(y|x)]$, the task-specific MI constrained to a predictor class V.

FID (Fréchet Inception Distance):

$W_2^2(N(\mu_{real}, \Sigma_{real}), N(\mu_{gen}, \Sigma_{gen}))$ between Inception-v3 feature distributions of real and generated images.

DP-SGD:

Differentially private SGD that clips per-example gradients to norm C and adds Gaussian noise $N(0, \sigma^2 C^2 I)$ at each step.

Natural gradient:

$\tilde{\nabla}_w L = I_F(w)^{-1} \nabla_w L$, the reparametrisation-invariant gradient on the statistical manifold.

Effective dimensionality:

$d_{eff}(w^*) = \mathrm{tr}(I_F(w^*)) / \lambda_{max}(I_F(w^*))$, the number of weight directions strongly constrained by the data.

Compression phase:

The hypothesised second phase of deep network training in the Tishby-Schwartz-Ziv framework, during which $I(X;T^\ell)$ decreases. Subsequently shown to be an artefact of binning estimators and saturating activations.

Phase transition (IB):

The bifurcation of the IB optimal encoder at a critical β_k, where the number of active codewords increases by 1.

Stochastic complexity:

$SC(x^n|M) = -\log p_{NML}(x^n) = -\log p(x^n|\hat{\theta}) + \log C_n(M)$, the MDL code length for x^n under model class M.

Parametric complexity:

$\log C_n(M) = (d/2) \log(n/(2\pi)) + \log \int \sqrt{\det I(\theta)}\, d\theta$, the model complexity penalty in MDL.

Bits-back coding:

The coding scheme where the receiver uses the approximate posterior $q(z|x)$ to 'save' $H(q(z|x))$ bits, giving a net communication cost of $D_{KL}(q(z|x) \,/\!/\, p(z))$ for the latent variable z.

SAM (Sharpness-Aware Minimisation):

Training algorithm that minimises $L_{SAM}(w) = \max_{\{ \| \varepsilon \| \le \rho \}} L(w+\varepsilon)$, seeking flat minima with good PAC-Bayes bounds.

Appendix D

Extended Proofs and Technical Derivations

D.1 Complete Proof of the IB Self-Consistent Equations

We provide the complete derivation of the IB self-consistent equations (Theorem 1.1), filling in all steps skipped in the main text.

Setup: we wish to minimise the IB functional $F_\beta[p(t|x)] = I(X;T) - \beta\, I(T;Y)$ subject to the constraint $\int p(t|x)\, dt = 1$ for all x. We use the functional derivative approach.

Step 1: expand the IB functional in terms of $p(t|x)$.

$I(X;T) = \iint p(x,t) \log(p(t|x)/p(t))\, dx\, dt$ where $p(t) = \int p(t|x)\, p(x)\, dx$.

$I(T;Y) = \iint p(t,y) \log(p(t|y)/p(t))\, dt\, dy = \iint p(y)\, p(t|y) \log(p(y|t)/p(y))\, dt\, dy$.

Since $p(y|t) = \int p(y|x)\, p(x|t)\, p(x)\, dx \,/\, p(t) = \int p(y|x)\, p(t|x)\, p(x)\, dx \,/\, p(t)$, the decoder $p(y|t)$ is a functional of $p(t|x)$.

Step 2: compute the functional derivative of $I(X;T)$ with respect to $p(t|x)$.

$\delta I(X;T)/\delta p(t|x) = p(x)\, [\log p(t|x) - \log p(t)] + p(x) - p(x) \cdot (\delta \log p(t))/\delta p(t|x) \cdot p(t)$.

The term involving $\delta p(t)/\delta p(t|x) = p(x)$ gives: $\delta(p(x) \log p(t))/\delta p(t|x) = p(x)^2 \,/\, p(t)$. Integrating out x in the stationarity condition, these terms cancel after applying the normalisation constraint.

The simplified result is: $\delta I(X;T)/\delta p(t|x) = p(x)\, [\log p(t|x)/p(t) + 1]$.

Step 3: compute the functional derivative of $I(T;Y)$ with respect to $p(t|x)$.

This is more involved because $p(y|t)$ depends on $p(t|x)$ through the Bayes formula. We use the chain rule: $\delta I(T;Y)/\delta p(t|x) = \partial I/\partial p(t|x)|_{p(y|t) \text{ fixed}} + \partial I/\partial p(y|t) \cdot \delta p(y|t)/\delta p(t|x)$.

The first term gives: $\partial I(T;Y)/\partial p(t|x)|_{p(y|t) \text{ fixed}} = -p(x) \sum_y p(y|x) \log p(y|t)$.

The second term: $\partial I(T;Y)/\partial p(y|t) = 0$ at the optimal $p^*(y|t)$ (by the envelope theorem — the decoder is self-consistently optimal). Therefore $\delta I(T;Y)/\delta p(t|x) = -p(x)\, D_{KL}(p(y|x) \,/\!/\, p(y|t))$.

Step 4: stationarity condition. Setting $\delta F_\beta/\delta p(t|x) = \lambda(x)$ (Lagrange multiplier for normalisation):

$$p(x)\, [\log p(t|x)/p(t) + 1] + \beta\, p(x)\, D_{KL}(p(y|x) \,/\!/\, p(y|t)) = \lambda(x).$$

Dividing by $p(x)$ and rearranging: $\log p(t|x) = \log p(t) - \beta\, D_{KL}(p(y|x) \,/\!/\, p(y|t)) + \lambda(x)/p(x) - 1$. Exponentiating: $p^*(t|x) \propto p(t) \exp(-\beta\, D_{KL}(p(y|x) \,/\!/\, p(y|t)))$, which is equation (E1). □

D.2 Proof of the PAC-Bayes Change-of-Measure Lemma

We prove the change-of-measure lemma: for any distributions P, Q over H and measurable $\phi : H \to \mathbb{R}$:

$$E_{h\sim Q}[\phi(h)] \le D_{KL}(Q \,\|\, P) + \log E_{h\sim P}[e^{\phi(h)}].$$

Proof. We use the Donsker-Varadhan variational representation of KL divergence:

$$D_{KL}(Q \,\|\, P) = \sup_{f} \{E_Q[f] - \log E_P[e^f]\},$$

where the supremum is over all measurable $f : H \to \mathbb{R}$. Setting $f = \phi$ in this variational formula gives:

$$D_{KL}(Q \,\|\, P) \ge E_Q[\phi] - \log E_P[e^\phi].$$

Rearranging: $E_Q[\phi] \le D_{KL}(Q \,\|\, P) + \log E_P[e^\phi]$. □

Alternative proof via Young's inequality. For any random variable Z and constant $c > 0$: $E_Q[Z] \le (1/c) \log E_P[e^{cZ}] + (1/c) D_{KL}(Q \,\|\, P)$. Setting $c = 1$ and $Z = \phi(h)$ gives the lemma directly. The Young's inequality approach also immediately gives the generalised Catoni bound by optimising c.

The change-of-measure lemma is a consequence of the Gibbs variational principle in statistical mechanics: the free energy $F = -(1/\beta) \log E_P[e^{-\beta E}]$ satisfies $F \le E_Q[E] + (1/\beta) D_{KL}(Q \,\|\, P)$ for any Q, with equality for the Gibbs distribution $Q^*(h) \propto P(h)\, e^{-\beta E(h)}$. Setting $E(h) = -\phi(h)/\beta$ gives the lemma.

D.3 Derivation of the Catoni Bound

We prove Theorem 2.3 (Catoni PAC-Bayes Bound). The key lemma is:

Lemma (Catoni 2007): for $Z \in [0,1]$ with $\mathbb{E}[Z] = \mu$ and $\lambda > 0$:

$$E[\log(1 - Z + Z\, e^{\lambda/n})] \le \lambda\, \mu / n.$$

Proof of lemma: by convexity of $\log(1 - z + z\, e^{\lambda/n})$ in $z \in [0,1]$, the function is bounded above by its tangent at $z = 0$: $\log(1 - z + z\, e^{\lambda/n}) \le (e^{\lambda/n} - 1)\, z$. Taking expectations: $\mathbb{E}[\log(1 - Z + Z\, e^{\lambda/n})] \le (e^{\lambda/n} - 1)\, \mu \le \lambda\mu/n$ (using $e^x - 1 \le ex$ for $x \ge 0$). □

Main proof. For each h, by Hoeffding's inequality over the training set S:

$$P(E_{h\sim P}[e^{n\lambda(L(h) - \hat{L}_S(h))}] > e^{D_{KL}(Q \,\|\, P) + \log(1/\delta)}) < \delta.$$

Combining with the change-of-measure lemma and the Catoni correction factor $(1/(1-e^{-\lambda/n}))$, we get the Catoni bound. The derivation involves: (1) applying the change-of-measure lemma to $\phi(h) = n\lambda(L(h) - \hat{L}(h))$; (2) bounding $E_P[e^\phi]$ using the Catoni lemma; (3) inverting to get $L(Q) \le f(\hat{L}(Q), D_{KL}, \lambda, n, \delta)$. □

D.4 Proof of the Blahut-Arimoto Convergence Theorem

We prove that the Blahut-Arimoto algorithm (Theorem 6.1) decreases the Lagrangian $\mathcal{L}_s = I(X;\hat{X}) + s \cdot \mathbb{E}[d(X,\hat{X})]$ at each step.

The BA algorithm alternates between two updates: E-step: $q(\hat{x}) \leftarrow \sum_x p(x)\, p(\hat{x}|x)$; M-step: $p(\hat{x}|x) \leftarrow q(\hat{x})\, e^{s\, d(x,\hat{x})} / \sum_{\hat{x}'} q(\hat{x}')\, e^{s\, d(x,\hat{x}')}$.

Claim: the E-step decreases $I(X;\hat{X})$ without changing $E[d(X,\hat{X})]$. The M-step decreases $s \cdot E[d(X,\hat{X})]$ without increasing $I(X;\hat{X})$.

Proof of E-step claim. The marginal update $q_{new}(\hat{x}) = \sum_x p(x)\, p(\hat{x}|x)$ uses the current encoder $p(\hat{x}|x)$ but updates the marginal to be consistent. By the identity $I(X;\hat{X}) = \sum_x p(x)\, D_{KL}(p(\hat{x}|x) \,\|\, q(\hat{x}))$:

$$I_{new}(X;\hat{X}) = \sum_x p(x)\, D_{KL}(p(\hat{x}|x) \,\|\, q_{new}(\hat{x})) \le \sum_x p(x)\, D_{KL}(p(\hat{x}|x) \,\|\, q_{old}(\hat{x})) = I_{old}(X;\hat{X}),$$

where the inequality uses the fact that q_{new} is the marginal under $p(\hat{x}|x) \cdot p(x)$, which minimises the average KL divergence (by the information projection theorem).

Proof of M-step claim. The M-step update is precisely the Gibbs encoder at inverse temperature $|s|$: $p_{new}(\hat{x}|x) \propto q(\hat{x})\, e^{s\, d(x,\hat{x})}$. This encoder minimises $E[d(X,\hat{X})]$ subject to the constraint $I(X;\hat{X}) \le I_{old}(X;\hat{X})$ (by the IB optimality condition). Since p_{new} is the IB-optimal encoder at the current $q(\hat{x})$, it achieves minimum expected distortion, so $E_{new}[d] \le E_{old}[d]$. The total Lagrangian $L_s = I(X;\hat{X}) + s\, E[d(X,\hat{X})]$ decreases at each step. □

D.5 Proof of the Gaussian Rate-Distortion Theorem

We prove Theorem 6.3 (Shannon-Kolmogorov rate-distortion for Gaussian sources) using the water-filling argument.

Setup: $X \sim N(0, \Sigma)$ with orthogonal decomposition $X = \sum_i \sigma_i\, u_i\, Z_i$ where $Z_i \sim N(0,1)$ i.i.d. and u_i are the eigenvectors of Σ. The reconstruction $\hat{X} = \sum_i \hat{X}_i$ with $\hat{X}_i$ in the direction u_i.

The rate-distortion function for the i-th component ($Z_i \sim N(0, \sigma_i^2)$):

$$R_i(D_i) = \max\{0,\ (1/2)\, \log(\sigma_i^2 / D_i)\},$$

where $D_i = E[(X_i - \hat{X}_i)^2]$ is the component distortion.

The total rate $R(D) = \sum_i R_i(D_i)$ subject to $\sum_i D_i = D$. Minimising $R(D)$ over $D_i \ge 0$ subject to $\sum_i D_i = D$ using Lagrange multipliers:

$$\partial R / \partial D_i = (1/2) \cdot (-\sigma_i^2 / D_i^2) \cdot (1 / \sigma_i^2 / D_i) \cdot (1 / (D_i / \sigma_i^2)) = ?\ No.$$

More carefully: $R_i(D_i) = (1/2)\, \log(\sigma_i^2 / D_i)$ for $D_i \le \sigma_i^2$, and $R_i = 0$ for $D_i > \sigma_i^2$. So $\partial R_i / \partial D_i = -1/(2 D_i)$ for $D_i \le \sigma_i^2$, and 0 otherwise.

At the optimum, the Lagrange condition $\partial R_i / \partial D_i = \mu$ for all active components gives $D_i^* = -1/(2\mu) = \lambda$ for some water-filling level λ. For inactive components, $D_i^* = \sigma_i^2$ (no compression). The water-filling condition $\sum_i D_i^* = \sum_i \min(\sigma_i^2, \lambda) = D$ determines λ. The optimal rate is $R(D) = \sum_i (1/2)\, \max\{0, \log(\sigma_i^2 / \lambda)\}$. □

D.6 Proof of Assouad's Lemma

We prove Theorem 7.3 (Assouad's Lemma) in detail.

Setup: parameter space $\Theta = \{-1,+1\}^d$, with $M \sim \text{Uniform}(\Theta)$. For each $\theta \in \Theta$, we observe $Y \sim P_\theta$ (n i.i.d. observations). We estimate θ by $\hat{\theta} = \hat{\theta}(Y)$ and measure error by $d(\hat{\theta}, \theta) = \sum_j 1[\hat{\theta}_j \neq \theta_j]$ (Hamming distance).

Step 1: decompose the risk by coordinate. For each coordinate j, define $M^{(j)}_+ = \sum_{\theta: \theta_j=+1} P_\theta / 2^{d-1}$ and $M^{(j)}_- = \sum_{\theta: \theta_j=-1} P_\theta / 2^{d-1}$ (the marginal distributions over coordinate j).

The risk decomposes as:

$$\max_\theta E[d(\hat{\theta}, \theta)] \geq E_{M \sim \text{Unif}(\Theta)}[d(\hat{\theta}, M)] = \sum_j P(\hat{\theta}_j \neq M_j).$$

For each j, $P(\hat{\theta}_j \neq M_j) \geq P_e^{(j)}$ where $P_e^{(j)}$ is the error probability for estimating coordinate j from observations drawn from $M^{(j)}_+$ or $M^{(j)}_-$.

Step 2: apply Le Cam's lemma to each coordinate. By Le Cam (Theorem 7.2) applied to the two-hypothesis problem ($\theta_j = +1$ vs. $\theta_j = -1$):

$$P_e^{(j)} \geq (1/2)(1 - \| M^{(j)}_+ - M^{(j)}_- \|_{TV}).$$

By assumption, $D_{KL}(P_\theta \| P_{\theta^{\wedge}j}) \leq \beta$ for all θ (where $\theta^{\wedge}j$ flips coordinate j). Averaging over the 2^{d-1} values of $(\theta_1, \ldots, \theta_{j-1}, \theta_{j+1}, \ldots, \theta_d)$: $D_{KL}(M^{(j)}_+ \| M^{(j)}_-) \leq \beta$ (by convexity of KL). Using $TV \leq \sqrt{(D_{KL}/2)}$: $\| M^{(j)}_+ - M^{(j)}_- \|_{TV} \leq \sqrt{(\beta/2)}$. Therefore $P_e^{(j)} \geq (1/2)(1 - \sqrt{(\beta/2)})$.

Step 3: combine. The separation assumption gives: whenever $\hat{\theta}_j \neq M_j$, $d(\hat{\theta}, M) \geq \alpha$ (the hypotheses are at least α apart in the estimation metric). Therefore:

$$\max_\theta E[d(\hat{\theta}, \theta)] \geq \sum_j a \cdot P_e^{(j)} \geq d \cdot a \cdot (1/2)(1 - \sqrt{(\beta/2)}) = (ad/2)(1 - \sqrt{(\beta/2)}). \square$$

D.7 Proof that Neural Collapse Minimises IB at $\beta \to \infty$

We prove Proposition 4.2: in the neural collapse regime, $I(T^L; X) \approx \log C$ and $I(T^L; Y) \approx H(Y)$ for balanced C classes.

Neural collapse (Papyan et al. 2020): in the terminal phase of training, the last-layer features h^L_i for examples in class c converge to the class mean μ_c, and the class means form a simplex equiangular tight frame (ETF): $\langle \mu_c, \mu_{c'} \rangle = -1/(C-1)$ for $c \neq c'$ and $\| \mu_c \| = 1$.

In the collapse regime: for example i in class c, $h^L_i \approx \mu_c$ (with small within-class variance $\varepsilon \to 0$). Therefore $T^L = g(Y)$ where $g(c) = \mu_c$ is a deterministic function of the class label Y.

$I(T^L; X)$: since T^L is a deterministic function of Y, and Y takes C values each with probability $1/C$ (balanced):

$$H(T^L) = H(g(Y)) = H(Y) = \log C.$$

$$H(T^L \mid X) = H(g(Y) \mid X) = H(Y \mid X) = 0 \quad \textit{(since } Y = h(X) \textit{ for some function } h \textit{ in the collapse regime).}$$

$$I(T\hat{}L; X) = H(T\hat{}L) - H(T\hat{}L \mid X) = \log C - 0 = \log C. \checkmark$$

I(T^L; Y): since T^L = g(Y) is a bijection from Y to $\{\mu_1, \ldots, \mu_C\}$:

$$I(T\hat{}L; Y) = H(Y) - H(Y \mid T\hat{}L) = H(Y) - 0 = H(Y) = \log C. \checkmark$$

For unbalanced classes ($p_c \neq 1/C$), $H(Y) = -\sum_c p_c \log p_c < \log C$, but $I(T\hat{}L; Y) = H(Y)$ still holds. The collapse achieves maximum $I(T\hat{}L; Y)$ (perfect class information) with minimum $I(T\hat{}L; X)$ ($\log C$ bits, the minimum needed to distinguish C classes). This is the IB fixed point at $\beta \to \infty$. □

D.8 Proof of the Wasserstein Continuity Theorem

We prove Theorem 5.3: if $\sup_x \| f_w(x) - f_{w'}(x) \| \leq \varepsilon$, then $W_1(f_w\#p(x), f_{w'}\#p(x)) \leq \varepsilon$.

Proof. The Kantorovich-Rubinstein duality gives: $W_1(\mu, \nu) = \sup_{f: \mathrm{Lip}(f)\leq 1} |\int f \, d\mu - \int f \, d\nu|$. For $\mu = f_w\#p$ and $\nu = f_{w'}\#p$:

$$\begin{aligned} \left|\int g \, d\mu - \int g \, d\nu\right| &= \left|E_p[g(f_w(X)) - g(f_{w'}(X))]\right| \\ &\leq E_p[\,|g(f_w(X)) - g(f_{w'}(X))|\,] \\ &\leq E_p[\mathrm{Lip}(g) \cdot \| f_w(X) - f_{w'}(X) \|\,] \\ &\leq 1 \cdot \sup_x \| f_w(x) - f_{w'}(x) \| \leq \varepsilon. \end{aligned}$$

Taking the supremum over all 1-Lipschitz g gives $W_1(\mu, \nu) \leq \varepsilon$. □

This proof uses the Kantorovich-Rubinstein duality (which characterises W_1 as a supremum of expectations over 1-Lipschitz test functions) and the Lipschitz continuity of g. The proof extends to W_p for any $p \geq 1$ by Hölder's inequality, but the resulting bound is $W_p \leq \varepsilon$ (for W_1 coupling) rather than a tight W_p bound.

D.9 The Fisher Information Geometry of Exponential Families

Exponential families are the most important class of parametric distributions for information geometry. A d-dimensional exponential family is:

$$p(x \mid \eta) = h(x) \exp(\eta\hat{}T \, T(x) - A(\eta)),$$

where $\eta \in R\hat{}d$ is the natural parameter, $T(x) \in R\hat{}d$ is the sufficient statistic, $A(\eta) = \log \int h(x) \exp(\eta\hat{}T \, T(x)) \, dx$ is the log-partition function, and h(x) is the base measure.

The Fisher information matrix of an exponential family is the Hessian of the log-partition function:

$$I_F(\eta) = \nabla^2_\eta A(\eta) = \mathrm{Cov}_\eta[T(X)].$$

This is a profound fact: the Fisher information (which measures the curvature of the log-likelihood) equals the covariance of the sufficient statistics (which measures the spread of the data). For exponential families, the Fisher information and the data covariance are identical.

The mean parameter $\mu = E_\eta[T(X)] = \nabla_\eta A(\eta)$ is related to η by the Legendre transform: $A^*(\mu) = \eta^\top \mu - A(\eta)$ (the convex conjugate of A). The mapping $\eta \mapsto \mu$ is a diffeomorphism (1-1 smooth map) between the natural parameter space and the mean parameter space.

The e-flat (exponential) and m-flat (mixture) structures:

- The exponential family is e-flat: geodesics in the e-connection are straight lines in the η-coordinate system. The projection of a distribution P onto the exponential family M (under the m-connection) is the maximum likelihood estimate: $\operatorname{argmin}_{\eta} D_{KL}(P \,/\!/\, p_\eta) = \eta$ such that $\mu = E_P[T(X)]$.
- The exponential family is m-flat: geodesics in the m-connection are straight lines in the μ-coordinate system. The projection onto the exponential family (under the e-connection) is the minimum KL divergence estimate: $\operatorname{argmin}_\eta D_{KL}(p_\eta \,/\!/\, P)$.

The Pythagorean theorem (Theorem 10.1) applies to exponential families: if M_e is the exponential family and P is any distribution, the e-projection $P^* = \operatorname{argmin}_{q \in M_e} D_{KL}(P \,/\!/\, q)$ satisfies $D_{KL}(P \,/\!/\, q) = D_{KL}(P \,/\!/\, P^*) + D_{KL}(P^* \,/\!/\, q)$ for all $q \in M_e$.

D.10 The NTK Kernel for Fully-Connected Networks

The Neural Tangent Kernel (NTK) for a fully-connected network with L layers, width n (neurons per layer), activation σ, and weights initialised as $w_{ij} \sim N(0, 2/n)$:

$$K^{(L)}(x, x') = K^{(L-1)}(x, x') \cdot \Sigma^{(L)}(x, x'),$$

where the recursion starts with $K^{(0)}(x, x') = x^\top x'$ and the kernel amplification at each layer is:

$$\Sigma^{(\ell)}(x, x') = E_{(u,v) \sim N(0, \left[\left[K^{(\ell-1)}(x,x), K^{(\ell-1)}(x,x')\right],\left[K^{(\ell-1)}(x',x), K^{(\ell-1)}(x',x')\right]\right])}[\sigma(u)\sigma(v)].$$

For ReLU activations $\sigma(z) = \max(0, z)$, the kernel amplification has the closed form:

$$\Sigma^{(\ell)}(x,x') = \frac{1}{2\pi}(\sin\theta + (\pi - \theta)\cos\theta) \cdot \sqrt{K^{(\ell-1)}(x,x)\, K^{(\ell-1)}(x',x')},$$

where $\theta = \arccos\left(K^{(\ell-1)}(x,x') \,/\, \sqrt{K^{(\ell-1)}(x,x)\, K^{(\ell-1)}(x',x')}\right)$. The NTK depends only on the angle between the inputs, not their norms.

Properties of the NTK:

- Positive definiteness: the NTK $K^{(L)}(x,x')$ is positive definite (as a kernel function) for all L and all input distributions with $P(K^{(L-1)}(x,x') > 0) > 0$.
- Depth alignment: as L increases, the NTK concentrates on the angle between inputs: $K^{(L)}(x,x') \to c_L \cdot \cos(\theta(x,x'))^L / (\cos(0))^L = c_L$ for $x = x'$. The deep NTK is nearly degenerate (rank 1) for very deep networks.
- Width limit: as $n \to \infty$, the NTK converges to the deterministic kernel $K_\infty(x,x')$ defined by the recursion above. For finite n, the NTK is random (depends on the random initialisation) but concentrates around K_∞ as $n \to \infty$.

Appendix E

Connections to Physics and Statistical Mechanics

The information-theoretic framework developed in this volume has deep connections to statistical mechanics and thermodynamics. These connections are not merely analogical — they are mathematical equivalences that allow tools from physics to be applied to machine learning, and vice versa.

E.1 The Free Energy and Partition Function

In statistical mechanics, a system with energy function $E(s)$ (where s is the system state) at inverse temperature β has the partition function $Z(\beta) = \sum_s e^{-\beta E(s)}$ and the free energy $F(\beta) = -(1/\beta) \log Z(\beta)$. The Boltzmann distribution (Gibbs distribution) is $P^*(s) = e^{-\beta E(s)} / Z(\beta)$.

In Bayesian learning, a learning algorithm with loss function $\hat{L}(h)$ (where h is the hypothesis) and prior P has the partition function $Z_{PAC}(\beta) = E_{h\sim P}[e^{-\beta n \hat{L}(h)}]$ and the PAC-Bayes free energy $F_{PAC}(\beta) = -(1/\beta) \log Z_{PAC}(\beta)$. The Gibbs posterior is $Q^*(h) \propto P(h)\, e^{-\beta n \hat{L}(h)}$.

The correspondence is: state s $\leftrightarrow$ hypothesis h, energy $E(s)$ $\leftrightarrow$ training loss $n\hat{L}(h)$, temperature $1/\beta$ $\leftrightarrow$ regularisation strength, partition function Z $\leftrightarrow$ normalising constant of the Gibbs posterior, free energy F $\leftrightarrow$ negative log-evidence (marginal likelihood).

The free energy has a variational representation: $F(\beta) = \inf_Q \{E_Q[\hat{L}(h)] + (1/\beta) D_{KL}(Q \,/\!/\, P)\}$, where the infimum is achieved by Q^* (the Gibbs posterior). This is the Gibbs variational principle, and it is the physics analogue of the PAC-Bayes bound.

E.2 Phase Transitions in Learning Systems

The phase transitions of the IB (Theorem 1.2) have a direct physical interpretation: they are second-order phase transitions in the statistical mechanical system defined by the IB Lagrangian. The order parameter is the number of active codewords $|T^*|$, which jumps discontinuously at each critical β_k.

The IB phase transitions are analogous to magnetic phase transitions in the Ising model: below the critical temperature T_c (corresponding to $\beta < \beta_1$), the system is in the disordered phase (all codewords equally likely); above T_c ($\beta > \beta_1$), the system breaks symmetry and selects a preferred codeword structure.

The Fisher information matrix plays the role of the susceptibility in statistical mechanics: $I_F(\beta) = -\partial^2 F/\partial\beta^2$ measures the fluctuations of the order parameter (the encoder distribution) around the equilibrium. At a phase transition, the susceptibility diverges: $I_F(\beta_k) = +\infty$. This is why the IB curve is non-differentiable at the critical values β_k — the Fisher information diverges there.

Deep network training dynamics can also exhibit phase transitions. The 'grokking' phenomenon (Power et al. 2022) — where networks trained on small datasets suddenly learn to generalise after prolonged training — is a phase transition in the learning dynamics: the network transitions from a memorisation regime (high I(T;X), low I(T;Y)) to a generalisation regime (low I(T;X), high I(T;Y)) at a critical training step.

E.3 The Renormalisation Group and Multi-Scale Learning

The renormalisation group (RG) is a theoretical framework in physics for studying systems at multiple scales. It provides a systematic way to 'coarse-grain' a system, integrating out fine-grained degrees of freedom to obtain an effective description at larger scales.

The connection to deep learning: each layer of a deep network performs a coarse-graining operation. The layer maps fine-grained input features (pixels, raw activations) to coarser representations (edges, textures, objects). This coarse-graining is analogous to the RG transformation, which maps a lattice model at scale a to a model at scale 2a by averaging over pairs of lattice sites.

The fixed points of the RG correspond to scale-invariant systems — systems that look the same at all scales. In deep learning, the 'fixed points' of the layer-wise coarse-graining correspond to representations that are invariant to scale changes: a representation T^ℓ is a fixed point if I(X; T^{ℓ+1}) ≈ I(X; T^ℓ) — the next layer does not further compress the input.

The information-theoretic formulation of the RG (Lenggenhager et al. 2020) uses the conditional mutual information I(X_fine; X_coarse | X_RG) to measure how much information is lost in each RG step. A perfect RG transformation would have I(X_fine; X_coarse | X_RG) = 0: the coarse-grained representation captures all the information about the fine-grained features that is relevant at the larger scale.

This suggests a deep connection between the IB, the RG, and deep networks: all three are descriptions of multi-scale information processing, where fine-grained features are progressively compressed into coarser representations while retaining the information relevant at each scale. The IB curve is the information-theoretically optimal trade-off; the RG provides the physical intuition for why deep networks implement this trade-off hierarchically.

Appendix F

Advanced Topics in Information-Theoretic Deep Learning

F.1 The Information Bottleneck for Structured Prediction

Structured prediction tasks — sequence labelling, parsing, image segmentation — require predicting structured outputs Y where the components of Y are statistically dependent. The IB framework extends to these settings, but the structure of Y creates new challenges and opportunities.

For sequence labelling (e.g., named entity recognition, POS tagging), the target Y = (Y_1, …, Y_T) is a sequence of labels for each position in the input sequence X = (X_1, …, X_T). The IB objective becomes:

$$\min_{p(t|x)} I(X;T) \quad \text{subject to} \quad I(T;Y_1,\ldots,Y_T) \geq I_0.$$

The mutual information I(T; Y_1, …, Y_T) = I(T; Y_1) + I(T; $Y_2|Y_1$) + … + I(T; Y_T|Y_1,…,Y_{T-1}) decomposes by the chain rule. An efficient encoder must capture not only the marginal predictive information for each position but also the correlations between labels at different positions.

For CRF-based sequence labelling, the IB-optimal encoder would capture the features of X that are informative about the entire label sequence Y, including the pairwise transition probabilities p(Y_t|Y_{t-1}) which are typically modelled separately. The VIB approach for sequence labelling (Kang et al. 2019) applies the VIB objective independently at each position, which captures the marginal predictive information but ignores inter-label correlations.

For image segmentation, Y ∈ {0,1}^{H×W} is a binary mask where each pixel is labelled as object or background. The IB-optimal encoder would capture the spatial context needed to disambiguate ambiguous pixels — exactly the information captured by the receptive field of each encoder neuron. The IB framework provides a principled explanation for why large receptive fields are beneficial for segmentation: they allow the encoder to capture more spatial context, increasing I(T; Y) at the same compression rate I(X; T).

F.2 Information Bottleneck and Fairness

Fairness in machine learning requires that predictions should not discriminate based on protected attributes (race, gender, age). The IB framework provides a natural approach to fairness: compress the input X to remove information about the protected attribute A while retaining information about the target Y.

The fairness-constrained IB: let A be the protected attribute and Y the target. The fairness-constrained encoder minimises:

$$\min_{p(t|x)} I(X;T) \quad \text{subject to} \quad I(T;Y) \geq R_Y \quad \text{and} \quad I(T;A) \leq R_A.$$

The first constraint ensures the representation retains sufficient predictive information about Y. The second constraint limits the information about A in the representation. The Lagrangian is:

$$L_{\beta,\gamma}[p(t|x)] = I(X;T) - \beta\, I(T;Y) + \gamma\, I(T;A),$$

which penalises representations that are informative about A (through the γ I(T;A) term) while rewarding representations that are informative about Y (through the β I(T;Y) term). Setting $\gamma > 0$ implements an automatic fairness constraint.

The fairness-accuracy trade-off: increasing γ reduces I(T;A) but may also reduce I(T;Y) (if A and Y are correlated). The trade-off curve $\{(I(T;A), I(T;Y)) : \gamma \geq 0\}$ is a Pareto frontier analogous to the IB curve. The minimum accuracy loss for a given level of fairness is determined by the conditional mutual information I(T;Y|A) — the predictive information in X beyond what can be attributed to A.

This is the information-theoretic lower bound on the fairness-accuracy trade-off: if the protected attribute A explains a fraction ρ^2 of the variance in Y (the squared correlation), then any representation that removes all information about A (I(T;A) = 0) loses at least ρ^2 fraction of the predictive information $I(T;Y) \leq (1-\rho^2)\, I(X;Y)$.

Theorem F.1 (Fairness-Accuracy Trade-off).
For jointly Gaussian (X, Y, A) with partial correlation $\rho_{YA|X}$ (correlation between Y and A conditioned on X):
$I(T;Y) \leq I(X;Y) - \rho^2_{YA|X} \cdot I(T;A)$ *for any encoder $p(t|x)$.*
The fairness-accuracy Pareto frontier is linear: every unit reduction in I(T;A) costs at least $\rho^2_{YA|X}$ units of I(T;Y).

F.3 Online Learning and the Information-Theoretic Regret

Online learning is the sequential prediction problem: at each round t, the learner observes x_t, predicts $\hat{y}_t$, observes y_t, and incurs loss $\ell_t = \ell(\hat{y}_t, y_t)$. The cumulative regret relative to a hypothesis h is $\text{Regret}_n(h) = \sum_t \ell_t - \sum_t \ell(h(x_t), y_t)$. Information-theoretic analysis of regret connects to the MDL framework.

The Bayesian regret for the Gibbs algorithm at inverse temperature λ is: $\text{Regret}_n(\text{Gibbs}_\lambda, h) \leq D_{KL}(\delta_h \,/\!/\, P) / \lambda + O(n\lambda)$. Setting $\lambda = \sqrt{(D_{KL}(\delta_h \,/\!/\, P) / n)}$ gives optimal regret $O(\sqrt{(D_{KL}(\delta_h \,/\!/\, P) \cdot n)})$ — the PAC-Bayes regret bound.

The information-theoretic interpretation: the regret of the Gibbs algorithm is the description length of the comparator hypothesis h under the prior P (in bits per unit loss), divided by the temperature λ. A hypothesis that is 'close to the prior' ($D_{KL}(\delta_h \,/\!/\, P) = -\log P(h)$ is small) has low regret, while a hypothesis that is 'far from the prior' has high regret.

Adaptive algorithms: the information-theoretic regret bound can be tightened by using an adaptive prior that concentrates on the best hypothesis seen so far. Follow-the-Regularised-Leader (FTRL) with a changing prior achieves regret

$O(\sqrt{(\log(1/P(h^*)) \cdot n)})$ where h^* is the best hypothesis in hindsight. This is the optimal regret for the online learning problem.

The connection to MDL: the FTRL regret $O(\sqrt{(L_{code}(h^*) \cdot n)})$ (where $L_{code}(h^*) = \log(1/P(h^*))$ is the MDL code length of h^*) matches the MDL generalisation bound (Theorem 3.3) up to a $\sqrt{n}$ factor. Online learning (minimising cumulative regret) and batch learning (minimising generalisation gap) are connected by the information-theoretic framework: both are controlled by the MDL code length of the optimal hypothesis.

F.4 The Information-Theoretic View of Attention

The attention mechanism in transformer models (Vaswani et al. 2017) can be understood as a soft information routing mechanism. An attention head with query Q, key K, and value V computes:

$$\mathit{Attn}(Q, K, V) = \mathit{softmax}(QK^T / \sqrt{d_k})\, V,$$

where d_k is the key dimension. The attention weights $a_{ij} = \mathrm{softmax}(q_i^T k_j / \sqrt{d_k})$ determine how much information flows from position j to position i: $a_{ij} \approx 1$ means position i attends fully to position j; $a_{ij} \approx 0$ means no information flows.

Information-theoretic interpretation: the attention weights a_{ij} define a soft routing matrix that determines the flow of information through the attention layer. The attention entropy $H_h = -(1/L) \sum_i \sum_j a_{ij} \log a_{ij}$ measures the 'diffuseness' of information routing:

- High H_h (diffuse attention): information is aggregated from all positions, producing a representation that captures global context. Each output position receives contributions from all input positions.
- Low H_h (sharp attention): information is copied from a single source position. Each output position receives information from at most one input position.

The mutual information between the attention output T^ℓ and the input X is bounded by:

$$I_\sigma(X; T^\ell) \leq \sum_h H_h^{(\ell)} \cdot \log(d_h) + C(\sigma, d),$$

because the attention output is a mixture of value vectors, and the entropy of a mixture is at most the entropy of the mixing distribution ($\log(d_h)$) times the attention entropy H_h.

This bound has practical implications for attention head pruning: heads with low attention entropy H_h contribute little to the representational diversity of the layer (they simply copy information from one position) and can be pruned with minimal loss of MI.

The query-key inner product $q_i^T k_j / \sqrt{d_k}$ can be interpreted as an information-theoretic relevance score: it measures how 'related' query position i is to key position j. Positions with high relevance scores should share more information, which is implemented by the softmax weighting.

The scaled dot-product attention q_i^T k_j / √d_k is related to the log-likelihood ratio between the joint distribution p(q_i, k_j) and the product of marginals p(q_i) p(k_j): if queries and keys are jointly Gaussian with covariance Σ_{QK}, then q_i^T Σ_{QK}^{-1} k_j is the natural information-theoretic relevance score. The unscaled dot-product q_i^T k_j is an approximation to this score when Σ_{QK} ≈ (1/d_k) I (the queries and keys are approximately uncorrelated).

F.5 Information-Theoretic Analysis of In-Context Learning

In-context learning (ICL) — the ability of large language models to perform tasks by processing a few examples in the input context without gradient updates — is one of the most surprising capabilities of modern LLMs. The information-theoretic framework provides a principled approach to understanding ICL.

The ICL setup: the model f_θ is given a context $C = (x_1, y_1, \ldots, x_k, y_k, x_{k+1})$ and predicts $\hat{y}_{k+1} = f_\theta(C)$. The k demonstration examples (x_i, y_i) teach the model the task; the test example x_{k+1} requires applying the learned task.

Information-theoretic model of ICL: define the 'task hypothesis' H = (task function h : X → Y). ICL works by updating the model's implicit posterior over H given the demonstrations C:

$$P(H \mid C) \propto P(H) \cdot \prod_i P(y_i \mid x_i, H).$$

The ICL prediction is approximately the Bayesian model average: $\hat{y}_{k+1} \approx E_{H \sim P(H|C)}[h(x_{k+1})]$. This is the Gibbs classifier at inverse temperature $\beta \to \infty$ (deterministic Bayes-optimal prediction).

The information gained from k demonstrations is $I(H; y_{1:k} \mid x_{1:k})$ — the mutual information between the task hypothesis and the observed labels. For simple tasks (binary classification, linear regression), this is approximately $k \cdot I(H; y_1 \mid x_1)$ (each demonstration provides roughly the same amount of information about H).

The minimum number of demonstrations for reliable ICL: for the model to assign probability ≥ 1-ε to the correct hypothesis H*, we need $I(H; y_{1:k} \mid x_{1:k}) \geq \log(1/(\varepsilon P(H^*)))$. Setting $k = \log(1/P(H^*)) / I(H; y_1 \mid x_1)$ gives the minimum number of demonstrations needed. For rare tasks (small P(H*)), more demonstrations are needed; for common tasks (large P(H*)), fewer.

Theorem F.2 (ICL Sample Complexity).
For a task H drawn from a prior P(H) with self-information L(H) = log(1/P(H)), the minimum number of in-context demonstrations needed for ICL to achieve accuracy ≥ 1-ε is: $k^* = \Omega(L(H) / I(H; y_1 \mid x_1))$.
Tasks with high self-information (rare tasks) require more demonstrations. Tasks where each demonstration is highly informative (high $I(H; y_1 \mid x_1)$) require fewer demonstrations.

F.6 Information-Theoretic Analysis of Memorisation vs. Generalisation

A fundamental question in deep learning is: when does a model memorise training data (storing specific input-output pairs) vs. generalise (learning the underlying data-generating process)? The information-theoretic framework provides a precise characterisation.

Memorisation: a model f_w memorises training example (x_i, y_i) if I(w; (x_i, y_i)) ≈ I(w; y_i | x_i) + I(w; x_i) — the model has stored information about both the input x_i and the label y_i. In the CMI framework (Theorem 7.6), memorisation corresponds to high CMI(A) = I(A(S); U | S̃) — the model's output is sensitive to which partition of the supersample was used for training.

Generalisation: a model f_w generalises if it has learned the mapping h : X → Y such that I(w; y | x, h) ≈ 0 — given the true mapping h, the model provides no additional information about y beyond what h already provides. In this case, the model has 'abstracted away' the specific training examples and learned the underlying rule.

The memorisation-generalisation spectrum can be measured by the CMI:

- CMI = 0: the model's output does not depend on which training partition was used. The model has learned something that is independent of the specific training examples — pure generalisation.
- CMI = H(U) = n log 2: the model's output uniquely determines the training partition. The model has memorised the training set — pure memorisation.
- Intermediate CMI: the model has partially generalised and partially memorised.

The grokking phenomenon (Power et al. 2022) — where models trained on small subsets of modular arithmetic first memorise then suddenly generalise — can be understood through the CMI framework. During the memorisation phase, CMI is high (the model has memorised the training partition). At the grokking point, CMI suddenly drops to nearly 0 as the model discovers a generalising algorithm.

The information-theoretic characterisation of grokking: the model is trained on data with a regular structure (modular arithmetic). The memorising solution has high CMI and high complexity (it must store all training examples). The generalising solution has low CMI and low complexity (it implements a simple arithmetic rule). The transition from memorisation to generalisation occurs when the weight decay (MDL penalty) makes the memorising solution more expensive than the generalising solution: when MDL(memorising) > MDL(generalising), the model transitions to the generalising solution.

F.7 Information Theory and Neural Architecture Search

Neural Architecture Search (NAS) automatically searches for optimal network architectures by evaluating candidate architectures on a validation set. The information-theoretic framework provides principled criteria for NAS that go beyond simple validation accuracy.

MDL-based NAS: the MDL principle suggests selecting the architecture A that minimises the total description length: MDL(A, S) = $L_1(A) + L_2(S \mid A)$. Here $L_1(A)$ = code length of the architecture (encoding the number of layers, widths, connections, activation functions) and $L_2(S \mid A)$ = training loss of the best model in architecture A. The MDL-optimal architecture is neither too simple (underfitting, large L_2) nor too complex (overfitting, large L_1).

PAC-Bayes-based NAS: the PAC-Bayes bound suggests selecting the architecture that minimises D_KL(Q_A // P_A) + $\hat{L}$(Q_A), where Q_A is the posterior (trained weights) and P_A is the prior (initialisation) for architecture A. Architectures with smaller KL divergence between initialisation and trained weights generalise better. This is the 'trainability' criterion for NAS: architectures where SGD does not move the weights far from the initialisation are preferred.

Proposition F.1 (NAS Generalisation Bound).
For an architecture A with p_A parameters, prior $P_A = N(0, \sigma_0^2 I_{p_A})$, and trained weights w^*_A with $\|w^*_A\|_2 = R_A$:
$L(Q_A) \le \hat{L}(Q_A) + \sqrt{[(R_A^2/(2\sigma_0^2) + \log(2\sqrt{n}/\delta)) / (2n)]}$.
The NAS-optimal architecture minimises $\hat{L}(Q_A) + R_A^2/(4n\sigma_0^2)$ — validation loss plus a term proportional to the squared weight norm, not the number of parameters p_A.

This proposition explains a well-known empirical observation: larger architectures (more parameters) do not necessarily generalise worse if their trained weight norms are similar to smaller architectures. The relevant quantity for generalisation is $R_A = \|w^*_A\|_2$ (the distance from the origin in weight space) rather than p_A (the number of parameters).

The MI-based NAS criterion: select the architecture A that maximises $I_\sigma(T^{L_A}; Y) / I_\sigma(X; T^{1_A})$ — the ratio of task-relevant information retained at the output to the input information processed at the first layer. This criterion favours architectures that efficiently route information from input to output.

F.8 Information Bottleneck for Graph Neural Networks

Graph Neural Networks (GNNs) process data on graphs G = (V, E) where nodes $v \in V$ have features x_v and edges $(u,v) \in E$ have optional features e_{uv}. The IB framework applies to GNNs by treating the node features as the input X and the graph-level (or node-level) predictions as the target Y.

The message-passing framework for GNNs updates node representations iteratively:

$$h^{(\ell+1)}_v = f_\ell(h^{(\ell)}_v, AGG_{u \in N(v)}[g_\ell(h^{(\ell)}_u, e_{uv})]),$$

where N(v) is the neighbourhood of v, AGG is an aggregation function (sum, mean, max), and f_ℓ, g_ℓ are learned functions. The IB analysis of GNNs asks: how much information does the ℓ-th layer representation $h^{(\ell)}_v$ retain about the k-hop neighbourhood of v (input information) and about the target prediction (relevance information)?

The IB curve for GNNs depends on the graph topology. For star graphs (one central node connected to many leaf nodes), the central node's representation must compress the information from all leaf nodes while retaining the relevant information for prediction. For path graphs (linear sequence of nodes), the information must propagate over long distances, and the compression-relevance trade-off at each hop determines whether long-range information is preserved.

The Weisfeiler-Lehman (WL) hierarchy and information theory: the WL algorithm for graph isomorphism testing assigns node labels based on the multiset of neighbourhood labels, iteratively refining the labels over k rounds. The k-th round WL labels are sufficient statistics for k-hop neighbourhood isomorphism. From the IB perspective, the k-th round WL labels are the minimal sufficient statistics for the graph structure at hop distance k, with I(T^k_WL; G) decreasing with k as more local structure is summarised.

Proposition F.2 (GNN Information Bounds).
For a GNN with ℓ message-passing layers and a graph with maximum degree d_{max}, the mutual information between the ℓ-th layer representation and the original node feature is bounded by:
$I(h^{(\ell)}_v; X_v) \leq I(h^{(0)}_v; X_v) = H(h^{(0)}_v)$,
$I(h^{(\ell)}_v; X_{N^\ell(v)}) \leq \sum_{u \in N^\ell(v)} I(h^{(0)}_u; X_u) \leq d_{max}^\ell \cdot H(X_v)$,
where $N^\ell(v)$ is the ℓ-hop neighbourhood of v. The second bound shows that GNN representations can aggregate information from exponentially many nodes (up to d_{max}^ℓ) but are limited by the entropy of the individual node features.

Appendix G

Open Problems and Future Directions

This appendix surveys the most important open problems at the intersection of information theory and deep learning, providing context for future research directions.

G.1 Tight Information-Theoretic Bounds for Transformers

The information-theoretic bounds developed in this volume (PAC-Bayes, MDL, IZS) all require bounded or sub-Gaussian losses. Transformer language models trained with cross-entropy loss have potentially unbounded loss (when the model assigns very low probability to the true next token), making the standard bounds inapplicable.

The key challenge is that the cross-entropy loss $\ell(w, (x,y)) = -\log p_w(y|x)$ has tails that are heavier than sub-Gaussian: for a model that assigns probability p to the correct token, the loss is $\log(1/p) \in [0, \infty)$. For standard PAC-Bayes bounds, the requirement $\ell \in [0,1]$ is essential; extending to unbounded losses requires different concentration inequalities.

Partial progress: Alquier (2021) derived PAC-Bayes bounds for ϱ-mixing processes (extending beyond i.i.d. data) and for bounded α-moments of the loss. These bounds apply to language modelling when the loss has bounded p-th moments for some $p > 1$, but the resulting bounds are significantly weaker than the i.i.d. case.

Open question 1: derive tight PAC-Bayes bounds for transformer language models that account for the heavy tails of the cross-entropy loss. Specifically, is there a bound of the form $L(Q) \leq \hat{L}(Q) + f(D_KL(Q \,//\, P), n)$ that is non-vacuous for large language models?

Open question 2: what is the correct notion of 'prior' for language model fine-tuning? The pre-trained model defines a natural prior on the output distribution (and hence on the weights, via the likelihood). Can the PAC-Bayes framework be applied with this data-dependent prior to obtain tight generalisation bounds for fine-tuned language models?

G.2 Information Geometry of Self-Attention

The self-attention mechanism computes attention weights $a_\{ij\} = \text{softmax}(q_i^T k_j / \sqrt{d_k})$. These weights define a time-varying, input-dependent mixing matrix over the sequence positions. The information-geometric structure of this mixing is poorly understood.

From the information geometry perspective, the attention operation is an m-projection: it computes the mixture distribution $p(\text{output_i}) = \sum_j a_\{ij\}\ p(\text{value_j})$, where $p(\text{value_j})$ is the distribution of the j-th value vector. The attention weights $a_\{ij\}$ are the mixture coefficients.

The e-projection of the attention mixture (projecting onto the exponential family of attention weights) gives the most 'efficient' attention pattern — the one that uses the minimum description length for the attention weights while approximating the optimal mixture. This is related to the MDL approach to attention head pruning.

Open question: what is the information-geometric structure of self-attention? Specifically, does the attention mechanism implement an e-projection, m-projection, or something else? Can the information-geometric framework explain the observed specialisation of attention heads (syntactic, positional, global)?

G.3 The Information Theory of Emergent Capabilities

Large language models exhibit emergent capabilities — qualitatively new abilities that appear suddenly as model scale increases — such as multi-step reasoning, code generation, and in-context learning. The information-theoretic basis of emergence is poorly understood.

A preliminary information-theoretic model: emergence occurs when the model's effective description length $L(w) = -\log P(w)$ (under the Jeffreys prior) crosses a threshold that makes a new algorithmic capability 'worth learning'. Below the threshold, the cost of implementing the capability (in bits of description length) exceeds the benefit (reduction in per-token cross-entropy). Above the threshold, the benefit exceeds the cost.

This model predicts that emergent capabilities should appear at specific model sizes that are determined by the complexity of the capability and the information content of the training data. The 'phase transition' in capability corresponds to a phase transition in the MDL objective: the model switches from 'not implementing' to 'implementing' the capability when the MDL cost-benefit crosses zero.

Open question: can the information-theoretic framework predict the scale at which specific capabilities emerge? Specifically, if we model a capability C as requiring K_C bits of 'implementation overhead' and providing I_C bits of per-token cross-entropy reduction, when does the MDL criterion select C? The answer depends on the training data distribution and the model capacity, and could potentially be computed from first principles.

G.4 Non-Asymptotic Information Theory for Deep Learning

Most of the theoretical results in this volume are asymptotic: they describe the behaviour of learning algorithms as $n \to \infty$. In practice, n is finite (often $n < 10^6$), and the asymptotic bounds may not be tight. Developing non-asymptotic information-theoretic bounds is a major open problem.

Non-asymptotic IB: the IB self-consistent equations (Theorem 1.1) characterise the optimal encoder for infinite data. For finite data, the empirical joint distribution $\hat{p}(x,y)$ replaces $p(x,y)$, and the IB-optimal encoder for $\hat{p}$ may differ significantly from the

optimal encoder for p. The finite-sample correction to the IB optimal solution is an open problem.

Non-asymptotic MDL: the stochastic complexity $SC(x^n|M) = -\log p_{NML}(x^n)$ involves the parametric complexity $\log C_n(M) = (d/2)\log n + \text{constant}$, which is the leading-order asymptotic term. The exact finite-n parametric complexity is the sum $C_n(M) = \sum_{z^n} p(z^n|\hat{\theta}(z^n))$, which cannot be simplified for general model classes. Tight non-asymptotic bounds on $\log C_n(M)$ are known only for specific model classes (Gaussian, Bernoulli, Poisson).

Non-asymptotic rate-distortion: the classical rate-distortion theorem (Shannon 1959) is asymptotic (for block length $n \to \infty$). For finite block lengths, the rate-distortion function $R(D)$ is replaced by the non-asymptotic rate-distortion function $R(n, D, \varepsilon)$ — the minimum rate to achieve distortion $\leq D$ with probability $\geq 1-\varepsilon$ for block length n. The finite-blocklength rate-distortion function is bounded by the classical $R(D) \pm O(\sqrt{(\log n / n)})$, but the precise constants are not known for neural network encoder-decoder pairs.

G.5 Information Theory for Continual Learning

Continual learning (lifelong learning) requires a model to learn a sequence of tasks $T_1, T_2, \ldots$ without forgetting previously learned tasks. The information-theoretic framework provides a principled analysis of the trade-off between learning new tasks and retaining knowledge of old tasks.

Catastrophic forgetting from the information-theoretic perspective: when a model is fine-tuned on task T_2 after being trained on T_1, the weight update $\Delta w = w_{T_2} - w_{T_1}$ increases $I(w; S_{T_2})$ while decreasing $I(w; S_{T_1})$. Catastrophic forgetting occurs when $\|\Delta w\|$ is large (the model moves far from the T_1 optimum) and the Fisher information $I_F(w_{T_1})$ is large in the direction of Δw (the T_1 loss is sensitive to changes in that direction).

The Elastic Weight Consolidation (EWC, Kirkpatrick et al. 2017) regulariser adds a penalty $\lambda/2 \sum_j F_j(w_j - w^*_{T_1,j})^2$ where $F_j = I_F(w_{T_1})_{jj}$ is the diagonal Fisher information from task T_1. This penalises changes to weights that are important for T_1 (high Fisher information) while allowing free changes to weights that are unimportant for T_1.

Theorem F.3 (EWC as MDL for Continual Learning).
EWC with regularisation strength λ is the MDL-optimal continual learning algorithm under the Gaussian prior $P_{T_1} = N(w^*_{T_1}, (1/\lambda) I_F(w_{T_1})^{-1})$.
The MDL description length for the new weights under this prior is:
$L_1(w) = (\lambda/2) \sum_j F_j (w_j - w^*_{T_1,j})^2 = D_{KL}(\delta_w \,\|\, P_{T_1}) + \text{const.}$
Minimising the total description length $MDL(w, S_{T_2}) = L_1(w) + L_2(S_{T_2}|w)$ gives the EWC objective.

This theorem establishes EWC as a principled MDL algorithm: the prior P_{T_1} encodes the knowledge from task T_1 (the trained weights $w^*_{T_1}$ and their Fisher information), and the MDL principle selects the simplest explanation of the new task data S_{T_2} consistent with this prior.

The information-theoretic limit of continual learning: for a sequence of tasks $T_1, T_2, \ldots$ with independent optimal parameters $\theta^*_1, \theta^*_2, \ldots$, the minimum description length to remember all tasks is $\sum_k L_{code}(\theta^*_k)$ bits (the total description length of all task-optimal parameters). This is the fundamental lower bound on the memory required for continual learning. Current continual learning methods do not achieve this bound, and the gap represents an open research opportunity.

G.6 Quantum Information Theory and Deep Learning

Quantum information theory extends classical information theory to quantum systems, where the state is a density matrix ϱ rather than a probability distribution p. The von Neumann entropy $S(\varrho) = -\mathrm{tr}(\varrho \log \varrho)$ replaces Shannon entropy, and quantum mutual information $I(A;B)_\varrho = S(\varrho_A) + S(\varrho_B) - S(\varrho_{AB})$ replaces classical MI.

Connections to deep learning: while deep networks are classical systems (operating on real-valued tensors), quantum information theory has inspired several directions in deep learning research:

- Tensor networks: quantum many-body states are represented as networks of tensors contracted along bonds. Tensor network architectures for deep learning (e.g., tensor trains, tree tensor networks) are mathematically equivalent to certain types of neural networks and can be analysed using quantum entanglement measures.
- Quantum-inspired algorithms: classical algorithms inspired by quantum mechanics (quantum annealing, variational quantum eigensolvers) have been applied to neural network training. The connection to the Gibbs algorithm (which is a classical analogue of quantum annealing) suggests that quantum-inspired algorithms may provide better approximations to the Gibbs posterior.
- Holographic duality: the AdS/CFT correspondence in theoretical physics relates quantum gravity in anti-de Sitter space to a conformal field theory on the boundary. The information-theoretic aspects of this duality (Ryu-Takayanagi formula, quantum error correction) have inspired new approaches to understanding the geometry of deep network representations.

Open question: are there quantum information-theoretic bounds on the complexity of deep learning that are tighter than classical bounds? Specifically, can quantum communication complexity (which measures the quantum resources needed to compute a function distributedly) provide tighter bounds on the sample complexity of federated learning?

Bibliography

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 308–318.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. (2017). Deep variational information bottleneck. In International Conference on Learning Representations (ICLR).

Amari, S. (2016). Information Geometry and Its Applications. Applied Mathematical Sciences, Vol. 194. Springer.

Ansuini, A., Laio, A., Macke, J. H., and Zoccolan, D. (2019). Intrinsic dimension of data representations in deep neural networks. In Advances in Neural Information Processing Systems (NeurIPS).

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein GAN. In International Conference on Machine Learning (ICML).

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. (2018). Stronger generalisation bounds for deep nets via a compression approach. In International Conference on Machine Learning (ICML).

Assouad, P. (1983). Deux remarques sur l'estimation. Comptes rendus de l'Académie des sciences, Série I, 296(23), 1021–1024.

Barron, A. R. (1985). Logically smooth density estimation. PhD thesis, Stanford University.

Barron, A. R. and Cover, T. M. (1991). Minimum complexity density estimation. IEEE Transactions on Information Theory, 37(4), 1034–1054.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. (2017). Spectrally-normalised margin bounds for neural networks. In Advances in Neural Information Processing Systems (NeurIPS).

Belghazi, M. I., Baratin, A., Rajeswar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, R. D. (2018). MINE: Mutual information neural estimation. In International Conference on Machine Learning (ICML).

Belkin, M., Hsu, D., Ma, S., and Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias-variance trade-off. Proceedings of the National Academy of Sciences, 116(32), 15849–15854.

Blau, Y. and Michaeli, T. (2018). The perception-distortion tradeoff. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Brenier, Y. (1991). Polar factorization and monotone rearrangement of vector-valued functions. Communications on Pure and Applied Mathematics, 44(4), 375–417.

Catoni, O. (2007). PAC-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning. IMS Lecture Notes-Monograph Series, Vol. 56.

Chechik, G., Globerson, A., Tishby, N., and Weiss, Y. (2005). Information bottleneck for Gaussian variables. Journal of Machine Learning Research, 6, 165–188.

Cover, T. M. and Thomas, J. A. (2006). Elements of Information Theory, 2nd edition. Wiley-Interscience.

Dawid, A. P. (1984). Statistical theory: The prequential approach. Journal of the Royal Statistical Society, Series A, 147(2), 278–292.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real-NVP. In International Conference on Learning Representations (ICLR).

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In Theory of Cryptography Conference (TCC), pp. 265–284.

Dziugaite, G. K. and Roy, D. M. (2017). Computing nonvacuous generalisation guarantees for deep (stochastic) neural networks with many parameters. In Uncertainty in Artificial Intelligence (UAI).

Federer, H. (1959). Curvature measures. Transactions of the American Mathematical Society, 93(3), 418–491.

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. Philosophical Transactions of the Royal Society of London A, 222, 309–368.

Foret, P., Kleiner, A., Mobahi, H., and Neyshabur, B. (2021). Sharpness-aware minimisation for efficiently improving generalisation. In International Conference on Learning Representations (ICLR).

Friedman, N., Mosenzon, O., Slonim, N., and Tishby, N. (2001). Multivariate information bottleneck. In Uncertainty in Artificial Intelligence (UAI).

Goldfeld, Z. and Polyanskiy, Y. (2020). The information bottleneck problem and its applications in machine learning. IEEE Journal on Selected Areas in Information Theory, 1(1), 19–38.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In Advances in Neural Information Processing Systems (NeurIPS).

Gong, R., Liu, X., Jiang, S., Li, T., Hu, P., Lin, J., Yu, F., and Yan, J. (2019). Differentiable soft quantisation: Bridging full-precision and low bit neural networks. In IEEE International Conference on Computer Vision (ICCV).

Grassberger, P. and Procaccia, I. (1983). Measuring the strangeness of strange attractors. Physica D: Nonlinear Phenomena, 9(1–2), 189–208.

Hanneke, S. and Kpotufe, S. (2019). On the value of target data in transfer learning. In Advances in Neural Information Processing Systems (NeurIPS).

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). β-VAE: Learning basic visual concepts with a constrained variational framework. In International Conference on Learning Representations (ICLR).

Hinton, G. E. and van Camp, D. (1993). Keeping the neural networks simple by minimising the description length of the weights. In Proceedings of the 6th Annual Conference on Computational Learning Theory (COLT).

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Advances in Neural Information Processing Systems (NeurIPS).

Hochreiter, S. and Schmidhuber, J. (1997). Flat minima. Neural Computation, 9(1), 1–42.

Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalisation in neural networks. In Advances in Neural Information Processing Systems (NeurIPS).

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2017). On large-batch training for deep learning: Generalisation gap and sharp minima. In International Conference on Learning Representations (ICLR).

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimisation. In International Conference on Learning Representations (ICLR).

Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1×1 convolutions. In Advances in Neural Information Processing Systems (NeurIPS).

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. (2021). Variational diffusion models. In Advances in Neural Information Processing Systems (NeurIPS).

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In International Conference on Learning Representations (ICLR).

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. Problems of Information Transmission, 1(1), 1–7.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. (2019). Similarity of neural network representations revisited. In International Conference on Machine Learning (ICML).

Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. Physical Review E, 69(6), 066138.

Le Cam, L. (1973). Convergence of estimates under dimensionality restrictions. Annals of Statistics, 1(1), 38–53.

McAllester, D. A. (1999). Some PAC-Bayesian theorems. Machine Learning, 37(3), 355–363.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralised data. In Artificial Intelligence and Statistics (AISTATS).

Mei, S., Montanari, A., and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. Proceedings of the National Academy of Sciences, 115(33), E7665–E7671.

Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one? In Advances in Neural Information Processing Systems (NeurIPS).

Mironov, I. (2017). Rényi differential privacy. In 2017 IEEE Computer Security Foundations Symposium (CSF), pp. 263–275.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. (2017). Exploring generalisation in deep learning. In Advances in Neural Information Processing Systems (NeurIPS).

Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In International Conference on Machine Learning (ICML).

Papyan, V., Han, X., and Donoho, D. L. (2020). Prevalence of neural collapse during the terminal phase of deep learning training. Proceedings of the National Academy of Sciences, 117(40), 24652–24663.

Poole, B., Ozair, S., van den Oord, A., Alemi, A., and Tucker, G. (2019). On variational bounds of mutual information. In International Conference on Machine Learning (ICML).

Rissanen, J. (1978). Modeling by shortest data description. Automatica, 14(5), 465–471.

Rissanen, J. (1996). Fisher information and stochastic complexity. IEEE Transactions on Information Theory, 42(1), 40–47.

Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B., and Cox, D. D. (2018). On the information bottleneck theory of deep learning. In International Conference on Learning Representations (ICLR).

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. Annals of Statistics, 48(4), 1875–1897.

Seeger, M. (2002). PAC-Bayesian generalisation error bounds for Gaussian process classification. Journal of Machine Learning Research, 3, 233–269.

Shannon, C. E. (1948). A mathematical theory of communication. Bell System Technical Journal, 27(3), 379–423.

Shannon, C. E. (1959). Coding theorems for a discrete source with a fidelity criterion. IRE National Convention Record, Part 4, 142–163.

Solomonoff, R. J. (1964). A formal theory of inductive inference. Information and Control, 7(1), 1–22.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modelling through stochastic differential equations. In International Conference on Learning Representations (ICLR).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

Steinke, T. and Zakynthinou, L. (2020). Reasoning about generalisation via conditional mutual information. In Conference on Learning Theory (COLT).

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. (2020). What makes for good views for contrastive learning? In Advances in Neural Information Processing Systems (NeurIPS).

Tishby, N., Pereira, F. C. N., and Bialek, W. (1999). The information bottleneck method. In Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing.

Tishby, N. and Schwartz-Ziv, R. (2017). Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810.

van den Oord, A., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (NeurIPS).

Villani, C. (2009). Optimal Transport: Old and New. Springer.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analysing multi-head self-attention: Specialised heads do the heavy lifting, the rest can be pruned. In Annual Meeting of the Association for Computational Linguistics (ACL).

Wang, T. and Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In International Conference on Machine Learning (ICML).

Xu, J. and Raginsky, M. (2017). Information-theoretic analysis of generalisation capability of learning algorithms. In Advances in Neural Information Processing Systems (NeurIPS).

Xu, Y., Zhu, S., Ma, T., and Du, S. S. (2020). A theory of usable information under computational constraints. In International Conference on Learning Representations (ICLR).